

# Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b>	<b>5</b>
1.1	Γιατί να χρησιμοποιήσουμε Νευρωνικά Δίκτυα; . . . . .	5
1.2	Τι είναι ένα νευρωνικό δίκτυο; . . . . .	6
1.2.1	Τεχνητά Νευρωνικά Δίκτυα . . . . .	6
1.2.2	Βιολογικά Νευρωνικά Δίκτυα . . . . .	8
1.3	Που χρησιμοποιούνται τα Νευρωνικά Δίκτυα . . . . .	10
1.3.1	Επεξεργασία Σημάτων . . . . .	10
1.3.2	Αναγνώριση Προτύπων . . . . .	11
1.3.3	Ιατρική . . . . .	11
1.3.4	Αναπαραγωγή Ομιλίας . . . . .	11
1.3.5	Αναγνώριση Ομιλίας . . . . .	12
1.3.6	Επιχειρήσεις . . . . .	13
1.4	Πώς Χρησιμοποιούνται; . . . . .	13
1.4.1	Τυπικές Αρχιτεκτονικές . . . . .	13
1.4.2	Καθορίζοντας τα βάρη . . . . .	16
1.4.3	Συνηθισμένες Συναρτήσεις Ενεργοποίησης . . . . .	18
1.5	Σύντομη Ιστορική Αναδρομή . . . . .	22
1.5.1	Δεκαετία 1940: Η Αρχή . . . . .	22
1.5.2	Δεκαετίες 1950 και 1960: Η Πρώτη Χρυσή Εποχή των Νευρωνικών Δικτύων . . . . .	23
1.5.3	Δεκαετία 1970: Τα Ήσυχά Χρόνια . . . . .	25
1.5.4	Δεκαετία 1980: Ανανεωμένος Ενθουσιασμός . . . . .	26
1.6	Κανόνες Μάθησης . . . . .	28
1.6.1	Μάθηση με επίβλεψη . . . . .	28
1.6.2	Μάθηση χωρίς επίβλεψη . . . . .	39

<b>2</b>	<b>Δυναμικά Νευρωνικά Δίκτυα</b>	<b>59</b>
2.1	Γραμμικός Συσχετιστής . . . . .	59
2.1.1	Προσαρμογή με τον Κανόνα Hebb . . . . .	61
2.1.2	Pseudohebbian Προσαρμογή . . . . .	63
2.2	Δίκτυο Hopfield . . . . .	67
2.2.1	Βασικό Μοντέλο . . . . .	67
2.2.2	Συνάρτηση Ενέργειας . . . . .	69
2.2.3	Χωρητικότητα της Μνήμης . . . . .	73
2.2.4	Παράδειγμα Εφαρμογής ενός Hopfield . . . . .	74
2.3	Συνεχές Δίκτυο Hopfield . . . . .	74
2.3.1	Συνεχής Υπολογιστική Δυναμική . . . . .	75
2.3.2	Το Πρόβλημα του Περιπλανόμενου Πωλητή (TSP) . . . . .	78
2.4	Μηχανή Boltzmann . . . . .	83
2.4.1	Στοχαστική Υπολογιστική Δυναμική . . . . .	84
2.4.2	Simulated Annealing . . . . .	86
2.4.3	Κατάσταση Ισορροπίας . . . . .	87
2.4.4	Μάθηση Boltzmann . . . . .	89
2.4.5	Αλγόριθμος Μάθησης . . . . .	91
<b>3</b>	<b>CELLULAR NEURAL NETWORKS</b>	<b>95</b>
3.1	THE CNN PARADIGM . . . . .	98
3.1.1	Εισαγωγή . . . . .	98
3.1.2	Το απλό πλαίσιο λειτουργίας του CNN . . . . .	99
3.1.3	Ο γενικός ορισμός του CNN παραδείγματος . . . . .	104
3.1.4	Παραδείγματα Εφαρμογών . . . . .	105
3.2	CNNs ΔΙΑΚΡΙΤΟΥ ΧΡΟΝΟΥ (Discrete-Time CNNs) . . . . .	111
3.2.1	Εισαγωγή . . . . .	111
3.2.2	Αρχιτεκτονική . . . . .	112
3.2.3	Εφαρμογές . . . . .	116
3.3	CNNs WITH NON-LINEAR AND DELAY-TYPE TEMPLATE ELEMENTS AND NON-UNIFORM GRIDS . . . . .	122
3.3.1	Εισαγωγή . . . . .	122
3.3.2	Το γενικό πλαίσιο λειτουργίας του CNN με μη-γραμμικά και delay-type templates . . . . .	122
3.3.3	Ειδικές Περιπτώσεις . . . . .	129
3.3.4	Τα CNN ως αναλογικός πίνακας υπολογισμού . . . . .	134
3.3.5	Μερικές Ποιοτικές ιδιότητες . . . . .	135

3.4	VLSI IMPLEMENTATION . . . . .	137
3.4.1	Εισαγωγή . . . . .	137
3.4.2	Υποθέσεις . . . . .	137
3.4.3	Εναλλακτικές Τεχνικές Ελέγχου . . . . .	138
3.4.4	Έλεγχος Κελιών . . . . .	138
3.4.5	Αρχιτεκτονική Στρατηγική . . . . .	140
3.4.6	Συμπεράσματα . . . . .	141
3.5	ΕΦΑΡΜΟΓΕΣ . . . . .	143
3.5.1	A CNN Handwritten Recognizer . . . . .	143
3.5.2	Detecting Moving and Standing Objects Using CNNs . . . . .	159
3.5.3	Cellular Neural Network Design For Solving Spe- cific Image-Processing Problems . . . . .	178



# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Γιατί να χρησιμοποιήσουμε Νευρωνικά Δίκτυα;

Όσο οι υπολογιστές γίνονται όλο και πιο γρήγοροι, οι επιστήμονες προσπαθούν να χρησιμοποιήσουν τις μηχανές για εργασίες απλές για τον άνθρωπο. Βασισμένοι σε παραδείγματα και αφού έχουμε δεχτεί ώθηση από έναν "δάσκαλο", μπορούμε εύκολα να αναγνωρίσουμε το γράμμα Α ή να ξεχωρίσουμε το ένα αντικείμενο από το άλλο. Ο παραδοσιακός σειριακός, βασισμένος στη λογική, ψηφιακός υπολογισμός υπερτερεί σε πολλούς τομείς, άλλα έχει αποδειχτεί ανεπαρκής σε αρκετούς τύπους προβλημάτων.

Τα Νευρωνικά δίκτυα ενδιαφέρουν τους ερευνητές για διαφορετικούς λόγους. Οι ηλεκτρολόγοι μηχανικοί βρίσκουν πολλές εφαρμογές στην επεξεργασία σημάτων και στη θεωρία ελέγχου. Οι μηχανικοί υπολογιστών διεγείρονται από το ενδεχόμενο εφαρμογής τους σε hardware και στη ρομποτική. Οι επιστήμονες υπολογιστών τα χρησιμοποιούν για επίλυση δύσκολων προβλημάτων σε περιοχές όπως η τεχνητή νοημοσύνη και η αναγνώριση προτύπων, ενώ για τους μαθηματικούς τα νευρωνικά δίκτυα είναι ένα ισχυρό εργαλείο για προβλήματα όπου η σαφής μορφή μιας μαθηματικής σχέσης μεταξύ των μεταβλητών δεν είναι γνωστή.

Υπάρχουν αρκετά θέματα όσον αφορά τη φύση τους. Για παράδειγμα, είναι συγκεκριμένο μέρος hardware (πχ ένα chip VLSI κυκλώματος) ή πρόγραμμα; Τα Νευρωνικά Δίκτυα βασικά είναι μαθηματικά μοντέλα

επεξεργασίας πληροφορίας. Παρέχουν μεθόδους αρκετά διαφορετικές από τις μηχανές Turing ή τους υπολογιστές με αποθηκευμένα προγράμματα.

## 1.2 Τι είναι ένα νευρωνικό δίκτυο;

### 1.2.1 Τεχνητά Νευρωνικά Δίκτυα

Ένα τεχνητό νευρωνικό δίκτυο είναι ένα σύστημα επεξεργασίας πληροφορίας που έχει κοινά χαρακτηριστικά στην απόδοσή του με τα βιολογικά νευρωνικά δίκτυα. Έχουν αναπτυχθεί σαν γενικεύσεις μαθηματικών μοντέλων της ανθρώπινης γνώσης ή της βιολογίας και βασίζονται στις ακόλουθες υποθέσεις:

1. Η επεξεργασία της πληροφορίας γίνεται σε πολλά απλά στοιχεία που καλούνται νευρώνες
2. Τα σήματα περνούν ανάμεσα στους νευρώνες με συνδέσεις
3. Κάθε σύνδεση συσχετίζεται με ένα βάρος, που σε ένα τυπικό νευρωνικό δίκτυο, πολλαπλασιάζει το μεταφερόμενο σήμα
4. Κάθε νευρώνας εφαρμόζει μια συνάρτηση ενεργοποίησης (μη γραμμική συνήθως) στα inputs του δικτύου του για να προσδιορίσει το output σήμα του.

Ένα νευρωνικό δίκτυο χαρακτηρίζεται από (1) τον τρόπο των συνδέσεων ανάμεσα στους νευρώνες (*αρχιτεκτονική*), (2) τη μέθοδο προσδιορισμού των βαρών στις συνδέσεις του (*εκπαίδευση ή αλγόριθμος μάθησης*) και (3) τη *συνάρτηση ενεργοποίησης*.

Ένα νευρωνικό δίκτυο αποτελείται από ένα μεγάλο αριθμό από βασικά επεξεργάσιμα στοιχεία που καλούνται *νευρώνες, μονάδες, κελιά ή κόμβοι*. Κάθε νευρώνας συνδέεται με τους άλλους νευρώνες με συνδέσμους, καθένας από αυτούς συσχετίζεται με ένα βάρος. Τα βάρη αναπαριστούν την πληροφορία που χρησιμοποιείται από το δίκτυο για να λύσει το πρόβλημα. Τα νευρωνικά δίκτυα μπορούν να εφαρμοστούν σε μεγάλη ποικιλία προβλημάτων, όπως αποθήκευση και ανάκληση προτύπων

Σχήμα 1.1: Ένα απλό (τεχνητό) νευρωνικό δίκτυο

ή δεδομένων, ταξινόμηση προτύπων, ομαδοποίηση παρόμοιων προτύπων ή επίλυση σε προβλήματα βελτιστοποίησης.

Κάθε νευρώνας έχει μια εσωτερική κατάσταση, την *ενεργοποίηση* που είναι μια συνάρτηση των inputs που δέχεται. Τυπικά, ένας νευρώνας στέλνει την ενεργοποίηση ως σήμα σε άλλους νευρώνες. Κάθε νευρώνας μπορεί να στείλει μόνο ένα σήμα τη φορά, αν και το σήμα μπορεί να γνωστοποιηθεί και σε άλλους νευρώνες.

Για παράδειγμα, ας θεωρήσουμε το νευρώνα  $Y$  του σχήματος 1.1, που δέχεται inputs από τους νευρώνες  $X_1, X_2$  και  $X_3$ . Οι ενεργοποιήσεις (output σήματα) τους είναι  $x_1, x_2$  και  $x_3$ , αντίστοιχα. Τα βάρη των συνδέσεων από τους  $X_1, X_2$  και  $X_3$  στο νευρώνα  $Y$  είναι  $w_1, w_2$  και  $w_3$ , αντίστοιχα. Το input του δικτύου  $y_{in}$ , στο νευρώνα  $Y$  είναι το άθροισμα των σημάτων από τους  $X_1, X_2$  και  $X_3$ , δηλ.

$$y_{in} = w_1x_1 + w_2x_2 + w_3x_3$$

Η ενεργοποίηση  $y$  του νευρώνα  $Y$  δίνεται από κάποια συνάρτηση του input  $y = f(y_{in})$ , πχ μια σιγμοειδή

$$f(x) = \frac{1}{1 + \exp(-x)},$$

Αν υποθέσουμε ότι ο νευρώνας  $Y$  συνδέεται και με τους νευρώνες  $Z_1$  και  $Z_2$  με βάρη  $v_1$  και  $v_2$ , αντίστοιχα (σχήμα 1.2). Ο  $Y$  στέλνει το σήμα  $y$  σε καθένα από αυτούς. Οι τιμές που δέχονται όμως είναι διαφορετικές,

### Σχήμα 1.2: Ένα πολύ απλό νευρωνικό δίκτυο

αφού στο κάθε σήμα αντιστοιχεί διαφορετικό βάρος. Αν και το δίκτυο του σχήματος 1.2 είναι πολύ απλό, η παρουσία ενός κρυφού νευρώνα σε συνδυασμό με μια μη γραμμική συνάρτηση ενεργοποίησης, του δίνει τη δυνατότητα να επιλύσει πολύ περισσότερα προβλήματα από ένα δίκτυο με ένα μόνο input και output. Από την άλλη μεριά είναι πιο δύσκολο να εκπαιδεύσει κανείς (να βρεθούν δηλαδή οι βέλτιστες τιμές για τα βάρη) ένα δίκτυο με κρυφά επίπεδα.

#### 1.2.2 Βιολογικά Νευρωνικά Δίκτυα

Αρκετά χαρακτηριστικά των βιολογικών νευρωνικών δικτύων μπορούν να βοηθήσουν στην κατανόηση σημαντικών χαρακτηριστικών των τεχνητών νευρωνικών δικτύων. Ένα βιολογικό νευρωνικό δίκτυο αποτελείται από τρία συστατικά: τις διακλαδισμένες αποφύσεις νευρώνων (dendrites), το σώμα (soma) και τον νευράξονα (axon). Το σώμα (ή το κελί) αθροίζει τα εισαρχόμενα σήματα. Όταν ληφθεί ικανοποιητικό input το κελί πυροδοτείται, δηλαδή μεταδίδει ένα σήμα μέσω του νευράξονά του στα άλλα κελιά. Θεωρούμε ότι ένα κελί είτε πυροδοτείται είτε όχι σε κάθε στιγμή του χρόνου, επομένως τα μεταδιδόμενα σήματα μπορούν να θεωρηθούν σαν δυαδικά.

Ένας βιολογικός νευρώνας στη γενική μορφή του φαίνεται στο σχήμα 1.3, με δύο νευράξονες από άλλους δύο νευρώνες και αποφύσεις για άλλους δύο νευρώνες. Αρκετά σημαντικά χαρακτηριστικά των μονάδων των τεχνητών νευρωνικών δικτύων εμπνέονται από ιδιότητες των βιολο-



### Σχήμα 1.3: Βιολογικός Νευρώνας

γικών νευρώνων:

1. Η μονάδα επεξεργασίας δέχεται πολλά σήματα
2. Τα σήματα μπορούν να τροποποιηθούν από ένα βάρος στη νευρική σύναψη
3. Η μονάδα επεξεργασίας αθροίζει τα inputs
4. Κάτω από κάποιες συνθήκες (ικανοποιητικό input) ο νευρώνας εκπέμπει output
5. Το output ενός συγκεκριμένου νευρώνα μπορεί να πάει σε άλλους νευρώνες
6. Η επεξεργασία της πληροφορίας γίνεται τοπικά
7. Η μνήμη διανέμεται ως εξής:
  - Η μακροπρόθεσμη (long-term) μνήμη διανέμεται στις συνάψεις του νευρώνα ή στα βάρη
  - Η βραχυπρόθεσμη (short-term) μνήμη ανταποκρίνεται στα σήματα που στέλνονται από τους νευρώνες

8. Η δύναμη μιας σύναψης μπορεί να τροποποιηθεί με την εμπειρία

Όμως άλλο ένα σημαντικό χαρακτηριστικό που μοιράζονται τα τεχνητά με τα βιολογικά νευρωνικά δίκτυα είναι η *ικανότητα αντοχής στο σφάλμα*. Τα βιολογικά νευρωνικά δίκτυα είναι ανεκτικά στο σφάλμα κατά δύο έννοιες. Πρώτον, μπορούμε να αναγνωρίσουμε πολλά input σήματα που είναι κατά κάποιον τρόπο διαφορετικά από κάθε άλλο σήμα που έχουμε δει (όπως η ικανότητα να αναγνωρίσουμε σε μια φωτογραφία ένα πρόσωπο που δεν έχουμε δει ή να αναγνωρίσουμε ένα πρόσωπο μετά από πολύ μεγάλο χρονικό διάστημα). Δεύτερον, μπορούμε να ανεχτούμε φθορά στο ίδιο το νευρωνικό σύστημα. Τα ανθρώπινα όντα γεννιούνται έχοντας περίπου 100 δισεκατομμύρια νευρώνες. Οι περισσότεροι από αυτούς βρίσκονται στον εγκέφαλο, όπως και οι περισσότεροι από αυτούς δεν αντικαθιστούνται όταν πεθάνουν. Παρά τη συνεχή απώλεια νευρώνων, συνεχίζουμε να μαθαίνουμε. Ακόμα και σε περιπτώσεις που υπάρχει τραυματική απώλεια νευρώνων, άλλοι νευρώνες συχνά εκπαιδεύονται ώστε να αναλάβουν τις λειτουργίες των κατεστραμμένων νευρώνων. Με παρόμοιο τρόπο, τα τεχνητά νευρωνικά δίκτυα μπορούν να σχεδιαστούν έτσι ώστε να μην επηρεάζονται από μικρή φθορά του δικτύου και το δίκτυο να μπορεί να εκπαιδευτεί ξανά σε περιπτώσεις σημαντικής φθοράς (πχ απώλεια δεδομένων και κάποιων συνδέσεων).

## 1.3 Που χρησιμοποιούνται τα Νευρωνικά Δίκτυα

### 1.3.1 Επεξεργασία Σημάτων

Υπάρχουν πολλές εφαρμογές νευρωνικών δικτύων σε αυτήν την περιοχή. Μια από τις πρώτες εμπορικές εφαρμογές ήταν (και ακόμα είναι) να εξαφανίσουμε τα παράσιτα σε μια τηλεφωνική γραμμή. Το δίκτυο που χρησιμοποιήθηκε για αυτόν το σκοπό είναι της μορφής ADALINE. Η ιδέα της εξουδετέρωσης των παρασίτων είναι αρκετά απλή. Στο τέλος μιας γραμμής μεγάλης απόστασης, το εισερχόμενο σήμα εφαρμόζεται και στο τηλεφωνικό σύστημα και στο ADALINE. Η διαφορά ανάμεσα στα output είναι το σφάλμα και χρησιμοποιείται για τη ρύθμιση των

βαρών του ADALINE. Το ADALINE εκπαιδεύεται για να απομακρύνει τα παράσιτα (αντίλαλος) από το output σήμα του τηλεφωνικού κέντρου.

### 1.3.2 Αναγνώριση Προτύπων

Μια πιο συγκεκριμένη περιοχή στην οποία αναπτύσσονται πολλές εφαρμογές νευρωνικών δικτύων είναι η αυτόματη αναγνώριση χειρόγραφων χαρακτήρων (ψηφία ή γράμματα). Η μεγάλη ποικιλία σε μεγέθη, θέσεις και ύφος γραψίματος, κάνει το πρόβλημα αρκετά δύσκολο για τις παραδοσιακές τεχνικές. Είναι ένα καλό παράδειγμα επεξεργασίας πληροφορίας που ο άνθρωπος εκτελεί αρκετά εύκολα.

### 1.3.3 Ιατρική

Μια από τις εφαρμογές στην ιατρική αναπτύχθηκε στα μέσα της δεκαετίας του '80, γνωστή ως "Γιατρός στη στιγμή" ("Instant Physician"). Η ιδέα ήταν να εκπαιδεύσουμε μια αυτοσυσχετιζόμενη μνήμη ("Brain-State-in-a-Box") για να αποθηκεύσουμε μεγάλο αριθμό ιατρικών αρχείων, καθένα από τα οποία περιέχει πληροφορίες για συμπτώματα, διαγνώσεις και θεραπείες για κάθε περίπτωση. Μετά την εκπαίδευση, το δίκτυο μπορεί να παρουσιαστεί έχοντας σαν inputs ένα σύνολο συμπτωμάτων. Θα βρεί το πλήρες αποθηκευμένο πρότυπο που παρουσιάζει την "καλύτερη" διάγνωση και θεραπεία. Το δίκτυο αποδίδει αρκετά καλά, αν και η δομή του είναι πολύ απλή. Όταν ένα συγκεκριμένο σύνολο από συμπτώματα παρουσιάζεται συχνά στο **training set**, μαζί με μια μοναδική διάγνωση και θεραπεία, το δίκτυο θα δώσει συνήθως την ίδια διάγνωση και θεραπεία. Σε περιπτώσεις που υπάρχουν αμφιβολίες στα δεδομένα, το δίκτυο θα δώσει την πιο συνηθισμένη διάγνωση και θεραπεία, ενώ σε νέες περιπτώσεις, το δίκτυο θα δώσει μια θεραπεία που θα αντιστοιχεί στα συμπτώματα που έχει ξαναδεί, άσχετα με τα νέα που του παρουσιάζονται.

### 1.3.4 Αναπαραγωγή Ομιλίας

Το να μάθει κανείς να διαβάζει Αγγλικό κείμενο δυνατά, είναι αρκετά δύσκολο αφού η σωστή προφορά ενός γράμματος εξαρτάται από την

έκφραση που εμφανίζεται το γράμμα. Μια παραδοσιακή μέθοδος κατασκεύαζε ένα σύνολο από κανόνες για την καθιερωμένη προφορά, μαζί με κάποιες εξαιρέσεις. Μια από τις πιο διαδεδομένες εφαρμογές για υτό το πρόβλημα είναι το NETtalk, ένα νευρωνικό δίκτυο με πολλά επίπεδα. Η μόνη απαίτησή του είναι ένα σύνολο από παραδείγματα του γραπτού input, μαζί με τη σωστή προφορά. Το γραπτό input περιέχει το τρέχον γράμμα ομιλίας συν τρία γράμματα πριν και μετά από αυτό. Πρόσθετα σύμβολα χρησιμοποιούνται για να οριστεί το τέλος μιας λέξης ή ένα σημείο στίξης. Το δίκτυο εκπαιδεύεται χρησιμοποιώντας τις 1000 πιο συνηθισμένες αγγλικές λέξεις. Μετά την εκπαίδευση μπορεί να διαβάσει νέες λέξεις κάνοντας ελάχιστα λάθη και ο λόγος είναι πολύ κατανοητός.

Έχει ενδιαφέρον ότι το δίκτυο περνά από αρκετά διακριτά στάδια κατά τη διάρκεια της εκπαίδευσης. Μαθαίνει αρκετά γρήγορα να διακρίνει τα φωνήεντα από τα σύμφωνα, όμως κατά το πρώτο στάδιο χρησιμοποιεί το ίδιο φωνήεν για όλα τα φωνήεντα και το ίδιο σύμφωνο για όλα τα σύμφωνα, αντίστοιχα. Το αποτέλεσμα είναι μια φλυαρία. Το δεύτερο στάδιο αφορά τα σύνορα στα οποία τερματίζει μια λέξη. Μετά από μικρό αριθμό, περίπου 10 περασμάτων, από τα δεδομένα εκπαίδευσης το κείμενο είναι κατανοητό. Επομένως η απόκριση του δικτύου σαν διαδικασία είναι παρόμοια με την ανάπτυξη της ομιλίας στα μικρά παιδιά.

### 1.3.5 Αναγνώριση Ομιλίας

Έχει σημειωθεί μεγάλη πρόοδος τελευταία σε αυτόν τον τομέα. Αρκετά συστήματα έχουν ένα περιορισμένο λεξιλόγιο ή γραμματική ή απαιτούν επανεκπαίδευση για διαφορετικούς ομιλητές. Αρκετοί τύποι δικτύων έχουν χρησιμοποιηθεί για αναγνώριση ομιλίας, συνήθως δίκτυα με πολλά επίπεδα ή και αναδράσεις. Ένα από αυτά είναι και το δίκτυο που σχεδίασε ο Kohonen και χρησιμοποιούσε τα **self-organizing maps**. Τα output τοποθετούνται σε διδιάστατους πίνακες. Το input του δικτύου βασίζεται σε μικρά τμήματα (λίγων χιλιοστών του δευτερολέπτου) της κυματομορφής του λόγου. Καθώς το δίκτυο ομαδοποιεί παρόμοια inputs, οι ομάδες που σχηματίζονται τοποθετούνται έτσι ώστε διαφορετικά παραδείγματα του ίδιου φωνήματος τυχαίνουν στα output που είναι τοποθετημένα μαζί στον output πίνακα. Αφού τα input σήματα αντιστοιχηθούν σε περιοχές φωνημάτων, τα outputs μπορούν να συνδεθούν στο κατάλληλο πλήκτρο ενός δακτυλογράφου για να κατασκευάσουν τα

φωνήματα. Επειδή η αντιστοιχία μεταξύ φωνημάτων και γραπτού είναι πολύ συμμετρική στα Φινλανδικά (για τα οποία κατασκευάστηκε το δίκτυο), η ορθογραφία είναι συνήθως σωστή.

### 1.3.6 Επιχειρήσεις

Ένα παράδειγμα είναι νευρωνικό δίκτυο που έχει εκπαιδευτεί για να εκτιμήσει το ρίσκο της παράβασης σε ένα δάνειο. Είναι βασισμένο σε δεδομένα από μια βάση με 111.080 εφαρμογές, από τις οποίες οι 109.072 δεν είχαν ιστορικό παραπτώματος. Ένα σύνολο από 4.000 δείγματα επιλέχθηκαν από τη βάση δεδομένων. Αν και η παράβαση μπορεί να είναι αποτέλεσμα πολλών αιτιών και δεν επηρεάζεται από τη διαθέσιμη πληροφορία της εφαρμογής του δανείου, οι προβλέψεις ωστόσο είχαν σαν αποτέλεσμα 12% μείωση στις παραβάσεις των εφαρμογών.

## 1.4 Πώς Χρησιμοποιούνται;

Ας δούμε μερικά λειτουργικά χαρακτηριστικά των νευρωνικών δικτύων. Θα ασχοληθούμε κυρίως με τις αρχιτεκτονικές των δικτύων και τις μεθόδους που χρησιμοποιούνται για να ρυθμίσουμε τα βάρη (εκπαίδευση).

### 1.4.1 Τυπικές Αρχιτεκτονικές

Είναι βολικό να απεικονίσουμε τους νευρώνες σε επίπεδα. Τυπικά, οι νευρώνες του ίδιου επιπέδου έχουν ίδια συμπεριφορά. Σημαντικοί παράγοντες στον καθορισμό της συμπεριφοράς ενός δικτύου είναι η συνάρτηση ενεργοποίησης και το πρότυπο των συνδέσεων μέσω των οποίων στέλνει και λαμβάνει σήματα. Στο ίδιο επίπεδο, οι νευρώνες έχουν συνήθως την ίδια συνάρτηση ενεργοποίησης και την ίδια δομή συνδέσεων με τους άλλους νευρώνες. Σε πολλά νευρωνικά δίκτυα, οι νευρώνες ενός επιπέδου είτε είναι πλήρως συνδεδεμένοι μεταξύ τους ή καθόλου.

Η διεύθυνση των νευρώνων στα επίπεδα και η μορφή των συνδέσεων ανάμεσα στα επίπεδα αποτελούν την αρχιτεκτονική του δικτύου. Συχνά κατατάσσονται σε ενός ή πολλών επιπέδων δίκτυα. Στον καθορισμό των επιπέδων, τα input επίπεδα δεν υπολογίζονται σαν επίπεδο, επειδή δεν κάνουν υπολογισμούς. Αντίστοιχα, ο αριθμός των επιπέδων στο

#### Σχήμα 1.4: Ένα νευρωνικό δίκτυο ενός επιπέδου

δίκτυο μπορεί να καθοριστεί ως ο αριθμός των επιπέδων των σταθμικών αλληλοσυνδέσεων μεταξύ των νευρώνων.

Τα δίκτυα των σχημάτων (1.4) και (1.5) είναι παραδείγματα **feedforward** δικτύων, δίκτυα στα οποία τα σήματα έχουν εμπρόσθια ροή. Το πλήρως συνδεδεμένο ανταγωνιστικό (**competitive**) δίκτυο του σχήματος (1.6) είναι παράδειγμα περιοδικού δικτύου (**recurrent**), στο οποίο υπάρχουν κλειστοί βρόχοι από ένα νευρώνα στον εαυτό του (αναδράσεις).

#### Δίκτυο ενός επιπέδου

Αποτελείται από ένα μόνο επίπεδο σταθμικών συνδέσεων. Οι νευρώνες συνήθως χωρίζονται σε input και output, σε αυτούς δηλαδή που δέχονται σήματα από εξωτερικές πηγές και σε αυτούς που δίνουν την απάντηση του δικτύου. Σε ένα τυπικό παράδειγμα (σαν το σχήμα (1.4)), οι input νευρώνες είναι πλήρως συνδεδεμένοι με τους output αλλά όχι και με άλλους input, ενώ οι output δεν συνδέονται με άλλους νευρώνες.

Για ταξινόμηση προτύπων, κάθε output αντιστοιχεί σε συγκεκριμένη κατηγορία, στην οποία ανήκει ή όχι ένα input διάνυσμα. Για συσχέτισμό προτύπων, χρησιμοποιείται η ίδια αρχιτεκτονική, απλώς το συνολικό πρότυπο των output σημάτων δίνει την απάντηση του δικτύου που σχε-

Σχήμα 1.5: Ένα νευρωνικό δίκτυο πολλών επιπέδων

Σχήμα 1.6: Ένα νευρωνικό δίκτυο με αναδράσεις

τίζεται με το input σήμα.

Πιο πολύπλοκα προβλήματα αντιστοιχίας όμως, απαιτούν ένα δίκτυο με κρυφά επίπεδα. Μπορεί να πρόκειται πάλι για προβλήματα ταξινόμησης ή συσχέτισης, ο τύπος του προβλήματος επηρεάζει την επιλογή της αρχιτεκτονικής, δεν την καθορίζει.

### **Δίκτυο πολλών επιπέδων**

Πρόκειται για δίκτυο με ένα ή περισσότερα επίπεδα νευρώνων μεταξύ των input και output νευρώνων (κρυφά επίπεδα). Προορίζονται για πιο πολύπλοκα προβλήματα, αλλά η εκπαίδευση τους μπορεί να είναι δύσκολη. Όμως, σε κάποιες περιπτώσεις, η εκπαίδευση μπορεί να είναι πιο πετυχημένη, επειδή είναι δυνατό να λυθεί το πρόβλημα που ένα δίκτυο ενός επιπέδου δεν μπορούσε να εκπαιδευτεί καν.

### **Ανταγωνιστικό δίκτυο**

Σχηματίζει ένα τμήμα αποτελούμενο από μεγάλο αριθμό νευρωνικών δικτύων. Τυπικά, οι αλληλοσυνδέσεις ανάμεσα στους νευρώνες δεν φαίνονται στην αρχιτεκτονική αυτών των δικτύων.

## **1.4.2 Καθορίζοντας τα βάρη**

Εκτός από την αρχιτεκτονική ενός δικτύου, ένας άλλος τρόπος διαχωρισμού των χαρακτηριστικών μεταξύ διαφορετικών δικτύων είναι και ο καθορισμός των βαρών (εκπαίδευση του δικτύου). Ξεχωρίζουμε κυρίως δύο τύπους, την εκπαίδευση με επίβλεψη και χωρίς επίβλεψη (*supervised* και *unsupervised* αντίστοιχα), υπάρχουν και δίκτυα των οποίων ο καθορισμός των βαρών δεν γίνεται με κάποια επαναλαμβανόμενη διαδικασία εκπαίδευσης.

Πολλά νευρωνικά δίκτυα εκπαιδεύονται να εκτελούν εργασίες που κατατάσσονται σε περιοχές αντιστοίχισης (*mapping*), συστοίχισης σε τομείς (*clustering*) και βελτιστοποίησης. Η ταξινόμηση προτύπων και η σύνδεση προτύπων μπορούν να θεωρηθούν ειδικές μορφές του πιο γενικού προβλήματος της αντιστοίχισης των input διανυσμάτων ή προτύπων στα αντίστοιχα outputs.



Υπήρξε μια αμφιβολία σχετικά με την περιγραφή των κατηγοριών εκπαίδευσης και κάποιοι συγγραφείς θεώρησαν απαραίτητο τον ορισμό μιας τρίτης κατηγορίας, της *self-supervised*. Επειδή λοιπόν υπάρχει μια χρήσιμη αντιστοιχία ανάμεσα στον τύπο προβλήματος που θέλουμε να επιλύσουμε και στον πιο κατάλληλο τύπο εκπαίδευσης, συνοψίζουμε τα βασικά χαρακτηριστικά τους.

### Εκπαίδευση με επίβλεψη

Στα πιθανώς πιο τυπικά παραδείγματα νευρωνικών δικτύων, η εκπαίδευση επιτυγχάνεται παρουσιάζοντας μια ακολουθία διανυσμάτων, ή προτύπων, καθένα από τα οποία συσχετίζεται με κάποιο από τα output διανύσματα. Τα βάρη τότε ρυθμίζονται σύμφωνα με κάποιον αλγόριθμο μάθησης. Αυτή η διαδικασία είναι γνωστή ως *εκπαίδευση με επίβλεψη*.

Κάποια από τα απλούστερα και παλαιότερα νευρωνικά δίκτυα έχουν σχεδιαστεί για την εκτέλεση της ταξινόμησης προτύπων, πχ να αποφασιστεί αν ένα input διάνυσμα ανήκει ή όχι σε κάποια κατηγορία. Σε αυτόν τον τύπο νευρωνικού δικτύου το output διάνυσμα είναι ένα δισθενές στοιχείο, για παράδειγμα έχει την τιμή +1 αν ανήκει στην κατηγορία, -1 διαφορετικά. Για πιο πολύπλοκα παραδείγματα ταξινόμησης μπορεί να χρησιμοποιηθεί ένα *backpropagation* δίκτυο.

Άλλος ένας τύπος προβλημάτων αντιστοίχισης είναι η σύνδεση προτύπων, όπου το επιθυμητό αποτέλεσμα είναι απλώς ένα "ΝΑΙ" ή "ΟΧΙ", και όχι κάποιο πρότυπο. Ένα δίκτυο που χρησιμοποιείται για να συσχετίσει ένα σύνολο διανυσμάτων με ένα σύνολο output διανυσμάτων είναι μια αυτοσυσχετιζόμενη μνήμη (*autoassociative memory*). Αν το δίκτυο επιτύχει, είναι μια αυτοσυσχετιζόμενη μνήμη, διαφορετικά είναι μια ετεροσυσχετιζόμενη μνήμη (*heteroassociative memory*). Μετά την εκπαίδευση, μια αυτοσυσχετιζόμενη μνήμη μπορεί να ανακαλέσει ένα αποθηκευμένο πρότυπο όταν της δοθεί ένα input διάνυσμα που είναι επαρκώς παρόμοιο με κάποιο από αυτά που εκπαιδεύτηκε. Για μη-γραμμική αντιστοίχιση από  $n$ -διάστατο χώρο σε  $m$ -διάστατο output χώρο χρησιμοποιούνται δίκτυα πολλών επιπέδων.

Τα πιο απλά δίκτυα ενός επιπέδου (δίκτυα ταξινόμησης και σύνδεσης προτύπων) χρησιμοποιούν τον κανόνα του *Hebb* ή τον *Delta* κανόνα. Τα *backpropagation* χρησιμοποιούν τον γενικευμένο *delta* κανόνα και ένα ακόμα παράδειγμα εκπαίδευσης με επίβλεψη είναι τα *LVQ* (*learning*

*vector quantization*).

### Εκπαίδευση Χωρίς επίβλεψη

Τα **self - organizing** νευρωνικά δίκτυα ομαδοποιούν παρόμοια input διανύσματα χωρίς τη χρήση δεδομένων εκπαίδευσης για να ορίσουν πώς πρέπει να μοιάζει το τυπικό στοιχείο μιας ομάδας ή σε ποιά ομάδα ανήκει κάθε διάνυσμα. Παρέχεται μια ακολουθία από input διανύσματα, αλλά όχι δεν καθορίζονται διανύσματα στόχου. Το δίκτυο τροποποιεί τα βάρη έτσι ώστε τα περισσότερο παρόμοια μεταξύ τους κατανέμονται στο ίδιο output (*cluster*). Τα πιο γνωστά self - organizing δίκτυα είναι τα *Kohonen self-organizing maps*.

### Δίκτυα σταθερών βαρών

Όμως κάποιοι άλλοι τύποι νευρωνικών δικτύων μπορούν να επιλύσουν προβλήματα βελτιστοποίησης με περιορισμούς. Αυτά τα δίκτυα μπορούν να δουλέψουν καλά σε προβλήματα που προκαλούν δυσκολίες στις παραδοσιακές τεχνικές, όπως προβλήματα με αντιθέμενους περιορισμούς (πχ, να μην μπορούν να ικανοποιηθούν όλοι οι περιορισμοί συγχρόνως). Συχνά, σε τέτοιες περιπτώσεις, η ευνοϊκότερη λύση είναι ικανοποιητική. Όταν σχεδιάζονται τα δίκτυα, τα βάρη διαμορφώνονται ώστε να αναπαριστούν τους περιορισμούς και την ποσότητα που πρόκειται να ελαχιστοποιηθεί ή να μεγιστοποιηθεί. Οι πιο γνωστές περιπτώσεις τέτοιων δικτύων είναι η μηχανή *Boltzmann* και το συνεχές *Hopfield* δίκτυο.

### 1.4.3 Συνηθισμένες Συναρτήσεις Ενεργοποίησης

Όπως αναφέρθηκε, η βασική λειτουργία ενός τεχνητού νευρωνικού δικτύου περιλαμβάνει και το άθροισμα των σταθμικών input προτύπων και την εφαρμογή τους σε κάποιο output, τη συνάρτηση ενεργοποίησης. Προκειμένου να επιτύχουμε τα πλεονεκτήματα των νευρωνικών δικτύων με πολλά επίπεδα, χρησιμοποιούμε μη γραμμικές συναρτήσεις ενεργοποίησης (αφού αν τροφοδοτήσουμε ένα πρότυπο, που περνά από δύο ή τρία επίπεδα, με μια γραμμική συνάρτηση δεν υπάρχει διαφορά αν χρησιμοποιήσουμε ένα δίκτυο ενός μόνο επιπέδου).

- Ταυτοτική συνάρτηση:

$$f(x) = x,$$

για κάθε  $x$

- **Binary step function** (με κατώφλι  $\theta$ ):

$$f(x) = \begin{cases} 1, & x \geq \theta \\ 0, & x < \theta \end{cases}$$

Οι σιγμοειδείς συναρτήσεις αποτελούν χρήσιμες συναρτήσεις ενεργοποίησης. Οι πιο συνηθισμένες είναι οι **logistic** και η συνάρτηση της υπερβολικής εφαπτομένης. Έχουν ιδιαίτερα πλεονεκτήματα για νευρωνικά δίκτυα με χρήση της backpropagation, επειδή η σχέση μεταξύ της τιμής προς την τιμή της παραγώγου σε ένα σημείο ελαττώνει σημαντικά τον υπολογιστικό κόπο κατά τη διάρκεια της εκπαίδευσης.

Η **logistic**, μια σιγμοειδής με πεδίο τιμών από 0 έως 1, χρησιμοποιείται για δίκτυα των οποίων τα outputs έχουν είτε δυαδικές τιμές ή τιμές στο διάστημα  $[0, 1]$ . Για να τονιστεί η σημασία του πεδίου τιμών της, θα την αποκαλούμε *δυαδική σιγμοειδή*.

- Δυαδική σιγμοειδής:

$$f(x) = \frac{1}{1 + \exp(-\sigma x)}$$

$$f'(x) = \sigma f(x)[1 - f(x)]$$

Η συνάρτηση μπορεί κλιμακωθεί για κάθε περιοχή τιμών που είναι σχετικές για το δοθέν πρόβλημα. Το πιο σύνηθες πεδίο είναι το  $[-1, 1]$ , τότε έχουμε τη *διπολική σιγμοειδή* (σχ. 1.10 με  $\sigma = 1$ ).

- Διπολική σιγμοειδής:

$$\begin{aligned} g(x) &= 2f(x) - 1 = \frac{2}{1 + \exp(-\sigma x)} - 1 \\ &= \frac{1 - \exp(-\sigma x)}{1 + \exp(-\sigma x)} \end{aligned}$$

Σχήμα 1.7: Η ταυτοτική συνάρτηση (γραμμική)

Η διπολική σιγμοειδής σχετίζεται με τη συνάρτηση της υπερβολικής εφαπτομένης, που χρησιμοποιείται επίσης συχνά ως συνάρτηση ενεργοποίησης για το ίδιο πεδίο. Ακολουθεί η αντιστοιχία μεταξύ τους, για  $\sigma = 1$ .

$$g(x) = \frac{1 - \exp(-x)}{1 + \exp(-x)}$$

Η υπερβολική εφαπτομένη είναι

$$\begin{aligned} h(x) &= \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \\ &= \frac{1 - \exp(-2x)}{1 + \exp(-2x)} \end{aligned}$$

Η παράγωγος της υπερβολικής εφαπτομένης είναι

$$h'(x) = [1 + h(x)][1 - h(x)]$$

Για δυαδικά δεδομένα (και όχι αντίστοιχα που παίρνουν συνεχείς τιμές στο  $[0,1]$ ), είναι προτιμότερο να μετατρέψουμε σε διπολική μορφή και να χρησιμοποιήσουμε τη διπολική σιγμοειδή ή τη συνάρτηση της υπερβολικής εφαπτομένης.

Σχήμα 1.8: Binary step function

Σχήμα 1.9: Δυαδική σιγμοειδής

Σχήμα 1.10: Διπολική σιγμοειδής

## 1.5 Σύντομη Ιστορική Αναδρομή

Ακολουθεί μια πολύ σύντομη ιστορική αναδρομή των νευρωνικών δικτύων, όσον αφορά την ανάπτυξη της αρχιτεκτονικής και τους αλγορίθμους που χρησιμοποιούν. Διακρίνεται η αλληλεπίδραση ανάμεσα στα βιολογικά μοντέλα και στην προσομοίωση τους σε hardware.

### 1.5.1 Δεκαετία 1940: Η Αρχή

#### McCulloch-Pitts νευρώνες

Οι Warren McCulloch και Walter Pitts σχεδίασαν αυτά που θεωρούνται ως τα πρώτα νευρωνικά δίκτυα (McCulloch and Pitts, 1941). Πρώτοι αυτοί κατάλαβαν ότι συνδυάζοντας πολλούς απλούς νευρώνες στα νευρωνικά συστήματα είχε ως αποτέλεσμα αυξημένη υπολογιστική δύναμη. Τα βάρη σε έναν McCulloch-Pitts νευρώνα καθορίζονται έτσι ώστε ο νευρώνας να εκτελεί μια συγκεκριμένη απλή λογική συνάρτηση, με διαφορετικούς νευρώνες να εκτελούν διαφορετικές συναρτήσεις. Οι νευρώνες συνεργάζονται σε ένα δίκτυο για να παράγουν οποιοδήποτε output που μπορεί να παρουσιαστεί σαν συνδυασμός λογικών συναρτήσεων. Δική τους ήταν και η ιδέα του κατώφλιού (*threshold*): αν το input του νευρώνα είναι μεγαλύτερο από το κατώφλι, τότε έχουμε εκπομπή σήματος. Κατά κύριο λόγο όμως οι νευρώνες των McCulloch-Pitts χρησιμοποιούνται

σαν λογικά κυκλώματα (Anderson - Rosenfeld, 1988). Αρκετά από τα προβλήματα που έθεσαν αποτελούν ακόμα σημαντικούς τομείς έρευνας, όπως η μετάφραση και η εναλλαγή σταθερών προτύπων για αναγνώριση (Pitts - McCulloch, 1947).

### **Μάθηση Hebb**

Ο Donald Hebb, ένας ψυχολόγος από το Πανεπιστήμιο McGill, σχεδίασε τον πρώτο κανόνα μάθησης (Hebb, 1949). Ο συλλογισμός του υποστήριζε πως αν δύο νευρώνες ήταν ενεργοί ταυτοχρόνως, η δύναμη της μεταξύ τους σύνδεσης πρέπει να αυξηθεί. Ακολούθησαν βελτιώσεις προκειμένου να επιτραπεί προσομοίωση σε υπολογιστή (Rochester, Holland, Haibt and Duda, 1956). Η ιδέα είναι πολύ κοντά στη μάθηση με πίνακα συσχέτισης που αναπτύχθηκε από τους Kohonen (1972) και Anderson (1972), μεταξύ άλλων. Μια ακόμη προσέγγιση υποστηρίζει ότι νευρώνες που είναι ταυτοχρόνως ανενεργοί, επίσης ενισχύουν το βάρος της μεταξύ τους σύνδεσης (McClelland - Rumelhart, 1988).

### **1.5.2 Δεκαετίες 1950 και 1960: Η Πρώτη Χρυσή Εποχή των Νευρωνικών Δικτύων**

Αν και σήμερα τα νευρωνικά δίκτυα αρκετά συχνά θεωρούνται ως εναλλακτική λύση (ή συμπλήρωμα) στους παραδοσιακούς ηλεκτρονικούς υπολογιστές, είναι ενδιαφέρον να σημειωθεί ότι ο John von Neumann, ο "πατέρας του σύγχρονου υπολογισμού", έδειξε έντονο ενδιαφέρον στην αναπαραγωγή σχεδιασμού του εγκεφάλου (von Neumann, 1958). Οι Johnson και Brown (1988) και Anderson και Rosenfeld (1988) συζητούν την αλληλεπίδραση ανάμεσα στον von Neumann και στους πρώτους ερευνητές νευρωνικών δικτύων, σαν τον Warren McCulloch.

### **Perceptrons**

Μαζί με αρκετούς ακόμα ερευνητές (Block 1962, Minsky - Papert 1988 (πρωτότυπη έκδοση 1969)), ο Frank Rosenblatt (1958, 1959, 1962) εισήγαγε και ανέπτυξε μια μεγάλη κλάση τεχνητών νευρωνικών δικτύων, τα *perceptrons*. Το πιο τυπικό perceptron αποτελείται από ένα input επίπεδο που συνδέεται με σταθερά βάρη με τους νευρώνες που συνεργάζεται.

Τα βάρη των μονοπατιών ήταν ρυθμιζόμενα. Ο κανόνας μάθησης χρησιμοποιεί μια επαναλαμβανόμενη ρύθμιση των βαρών πιο δυνατή από τον κανόνα του Hebb. Μπορεί να αποδειχτεί ότι η μάθηση του perceptron συγκλίνει στα σωστά βάρη, αν υπάρχουν βάρη που θα επιλύσουν το πρόβλημα σίγουρα. Η δουλειά του Rosenblatt (1962) περιγράφει πολλούς τύπους perceptrons. Όπως και οι νευρώνες των McCulloch και Pitts και του Hebb, τα perceptrons χρησιμοποιούν μια output συνάρτηση με κατώφλι.

Η αρχική επιτυχία των perceptrons προκάλεσε ενθουσιασμό. Όμως, η μαθηματική απόδειξη της σύγκλισης του επαναλαμβανόμενου κανόνα, υπό τις κατάλληλες υποθέσεις, ακολουθήθηκε από μια απόδειξη των περιορισμών σχετικά με το αν μπορεί ή όχι να μάθει το συγκεκριμένο δίκτυο [Minsky & Papert, 1969].

## ADALINE

Ο Bernard Widrow και ο μαθητής του Marcian Hoff (Widrow - Hoff, 1960), ανέπτυξαν ένα κανόνα μάθησης πολύ κοντά στον κανόνα του perceptron. Ο κανόνας του perceptron ρυθμίζει τα βάρη σε ένα νευρώνα όταν η απόκριση του νευρώνα είναι λανθασμένη (η απόκριση δηλώνει μια ταξινόμηση του input προτύπου). Ο κανόνας **delta** ρυθμίζει τα βάρη για να ελαττώσει τη διαφορά ανάμεσα στο input του δικτύου, το output και το επιθυμητό output του δικτύου. Αυτό αντιστοιχεί στη μικρότερη διαφορά τετραγώνων. Η ομοιότητα μεταξύ μοντέλων που αναπτύχθηκαν στη ψυχολογία από τον Rosenblatt και μοντέλων που αναπτύχθηκαν στην επιστήμη των μηχανικών από τους Widrow και Hoff, αποδεικνύει την πολυδιάστατη φύση των νευρωνικών δικτύων. Η, έστω και ελάχιστη, διαφορά στους κανόνες μάθησης, οδηγεί σε μια βελτιωμένη ικανότητα του δικτύου για γενίκευση (ανταπόκρισή του σε δεδομένα παρόμοια, αλλά όχι ίδια με αυτά στα οποία εκπαιδεύτηκε). Ο κανόνας των Widrow και Hoff για δίκτυα ενός επιπέδου αποτελεί προάγγελο της backpropagation για πολυ-επίπεδα δίκτυα. Πολλές φορές δίνεται στα δίκτυα του Widrow το όνομα *ADALINE*, που μεταφράζεται ως *ADaptive LInear NEuron* ή *ADaptive LINEar system*. Υπήρξαν πολλές ενδιαφέρουσες εφαρμογές των ADALINES (ή των MADALINES που ακολούθησαν), από προηγμένα συστήματα κεραίας, ως αναγνώριση προτύπων και συστήματα ελέγχου.



### 1.5.3 Δεκαετία 1970: Τα Ήσυχα Χρόνια

Παρά την επίδειξη των Minsky και Papert για τους περιορισμούς των perceptrons, η έρευνα για τα νευρωνικά δίκτυα συνεχίστηκε. Πολλοί από τους πρωταγωνιστές του χώρου άρχισαν την έρευνά τους τότε.

#### Kohonen

Η αρχική δουλειά του Teuvo Kohonen (1972), του Πανεπιστημίου Τεχνολογίας του Ελσίνκι, είχε σχέση με νευρωνικά δίκτυα που χρησιμοποιούνται σαν συσχετιζόμενες μνήμες. Η πιο πρόσφατη δουλειά του είναι η ανάπτυξη των **self-organizing feature maps** τα οποία χρησιμοποιούν μια τοπολογική δομή για τα **cluster units**. Αυτά τα δίκτυα έχουν εφαρμοστεί για αναγνώριση ομιλίας (σε Φινλανδική και Ιαπωνική γλώσσα) (Kohonen, Torkkola, Shozakai, Kangas and Venta, 1987, Kohonen, 1988), για την επίλυση του "Προβλήματος του Περιπλανώμενου Πωλητή" ("Travelling Salesman Problem") (Angeniol, Vaubois, and Le Texier, 1988) και σύνθεση μουσικής (Kohonen, 1989b).

#### Anderson

Ο James Anderson, του πανεπιστημίου Brown, επίσης ξεκίνησε την έρευνά του με συσχετιζόμενες μνήμες (Anderson, 1968, 1972). Ανέπτυξε αυτές τις ιδέες στο "**Brain-State-in-a-Box**" (Anderson, Silverstein, Ritz, and Jones, 1977), που περικλύπτει το γραμμικό output προηγούμενων μοντέλων προκειμένου να αποτρέψει το output από το να γίνει πολύ μεγάλο, καθώς οι επαναλήψεις στο δίκτυο συνεχίζονται μέχρι να βρεθεί σταθερή λύση (ή μνήμη). Ανάμεσα στις περιοχές των εφαρμογών του, συναντάμε διαγνώσεις στην ιατρική και εκμάθηση πινάκων πολλαπλασιασμού.

#### Grossberg

Ο Stephen Grossberg, με αρκετούς συναδέλφους του, είχε μια ιδιαίτερα παραγωγική καριέρα. Ο Klimasauskas (1989) έχει καταγράψει 146 δημοσιεύσεις του Grossberg από το 1967 ως το 1988. Η δουλειά του, κυρίως στα μαθηματικά και τη βιολογία, έγινε ιδιαίτερα γνωστή (Grossberg, 1976, 1980, 1982, 1987, 1988).

## Carpenter

Μαζί με τον Stephen Grossberg, ο Gail Carpenter ανέπτυξε μια θεωρία για **self-organizing** νευρωνικά δίκτυα, γνωστή και ως **adaptive resonance theory** (Carpenter - Grossberg, 1985, 1987a, 1987b, 1990). Γνωστά είναι ως ART1 (adaptive resonance theory nets για δυαδικά inputs) και ART2 (για συνεχή inputs).

### 1.5.4 Δεκαετία 1980: Ανανεωμένος Ενθουσιασμός

#### Backpropagation

Δύο από τους λόγους για τα "ήσυχια χρόνια" της δεκαετίας του '70 ήταν η αποτυχία των δικτύων με ένα μόνο layer για επίλυση απλών προβλημάτων (όπως η λογική συνάρτηση XOR) και η έλλειψη μιας γενικής μεθόδου εκπαίδευσης ενός δικτύου με πολλαπλά επίπεδα. Μια μέθοδος για διάδοση πληροφορίας του σφάλματος από τους output κόμβους προς τα πίσω, προς τους κρυμμένους, ανακαλύφθηκε τη δεκαετία του '70, αλλά δεν είχε μεγάλη απήχηση (Werbos, 1974). Αυτή η μέθοδος ανακαλύφθηκε ανεξάρτητα από τους Parker (1985) και LeCun (1986), πριν γίνει ευρέως γνωστή. Η δουλειά του Parker κίνησε το ενδιαφέρον του Parallel Distributed Processing Group, υπό την καθοδήγηση του David Rumelhart, του Πανεπιστημίου της Καλιφόρνια στο Σαν Ντιέγκο και του James McClelland, του Carnegie-Mellon, οι οποίοι τη βελτίωσαν και τη δημοσίευσαν (Rumelhart, Hinton and Williams, 1986a, 1986b; McClelland - Rumelhart, 1988).

#### Δίκτυα Hopfield

Ο διακεκριμένος φυσικός John Hopfield (California Institute of Technology), μαζί με τον David Tank, έναν ερευνητή του AT & T, ανέπτυξε μεγάλο αριθμό νευρωνικών δικτύων βασισμένων σε σταθερά βάρη και προσαρμόσιμες ενεργοποιήσεις [Hopfield, 1982, 1984; Hopfield - Tank 1985, 1986; Tank - Hopfield, 1987]. Αυτά τα δίκτυα μπορούν να εκτελούν καθήκοντα συσχετιζόμενης μνήμης και μπορούν να χρησιμοποιηθούν για επίλυση προβλημάτων με περιορισμούς, όπως το "Πρόβλημα του Περιπλανώμενου Πωλητή". Ένα άρθρο στο *Scientific American* (Tank - Hopfield, 1987), βοήθησε στην κίνηση γενικού ενδιαφέροντος για τα

νευρωνικά δίκτυα, όσο και το μήνυμα ενός φυσικού που κέρδισε βραβείο Nobel, ότι προκειμένου να κάνουμε τις μηχανές να κάνουν ό,τι κάνει ένας άνθρωπος, πρέπει να μελετήσουμε την ανθρώπινη γνώση.

### Μηχανή Boltzmann

Πολλοί ερευνητές έχουν ασχοληθεί με την ανάπτυξη μη-ντετερμινιστικών νευρωνικών δικτύων, δικτύων δηλαδή την οποίων τα βάρη ή οι ενεργοποιήσεις μεταβάλλονται με βάση μιας συνάρτησης πυκνότητας πιθανότητας (Kirkpatrick, Gelatt and Vecchi, 1983; Geman and Geman, 1984; Ackley, Hinton and Sejnowski, 1985; Szu and Hartley, 1987). Αυτά τα δίκτυα ενσωματώνουν ιδέες όπως η **Bayesian θεωρία απόφασης** και **Simulated Annealing**.

### Hardware Εφαρμογές

Ένας ακόμα λόγος για το ανανεωμένο ενδιαφέρον για τα νευρωνικά δίκτυα είναι και η βελτιωμένες υπολογιστικές δυνατότητες. Αναπτύχθηκαν οπτικά νευρωνικά δίκτυα (Farhat, Psaltis, Prata and Paek, 1985) και VLSI εφαρμογές (Sivilati, Mahowald and Mead, 1987).

Ο Carver Mead (California Institute of Technology), που μελέτησε επίσης και ανίχνευση της κίνησης, θεωρείται ο εφευρέτης στο σχεδιασμό microchips. Ο τιμημένος με Nobel Leon Cooper (Brown University) εισήγαγε ένα από τα πρώτα δίκτυα με πολλά επίπεδα, το *reduced coulomb energy network*. Οι Robert Hecht-Nielsen και Todd Gutschow ανέπτυξαν αρκετούς *neurocomputers* στην TRW, Inc. (1983-85). Χρηματοδότηση εξασφαλίστηκε από την DARPA (Defense Advanced Research Projects Agency) (Hecht-Nielsen, 1990).

## 1.6 Κανόνες Μάθησης

Μια από τις σημαντικότερες ικανότητες ενός νευρωνικού δικτύου είναι η ικανότητά του να μαθαίνει αλληλεπιδρώντας με το περιβάλλον του ή κάποια πηγή πληροφορίας. Η μάθηση επιτυγχάνεται μέσω μιας προσαρμόσιμης διαδικασίας, γνωστής ως *κανόνας μάθησης* ή *αλγόριθμος*, όπου τα βάρη του δικτύου ρυθμίζονται ώστε να βελτιώσουν ένα προκαθορισμένο μέτρο απόδοσης.

Στο περιβάλλον των νευρωνικών δικτύων, η διαδικασία της μάθησης μπορεί να θεωρηθεί σαν μια διαδικασία βελτιστοποίησης. Πιο συγκεκριμένα, μπορεί να θεωρηθεί σαν "έρευνα" σε έναν πολυδιάστατο παραμετρικό χώρο βαρών για κάποια λύση, που σταδιακά βελτιστοποιεί μια αντικειμενική συνάρτηση.

### 1.6.1 Μάθηση με επίβλεψη

Σε αυτήν την περίπτωση έχουμε διαχωρισμό σε κανόνες διόρθωσης σφάλματος και σε μεθόδους απότομης κλίσης. Θα καταλήξουμε στο συμπέρασμα ότι όλοι αυτοί οι κανόνες μπορούν να ελαχιστοποιήσουν μια κατάλληλη αντικειμενική συνάρτηση.

#### Κανόνες Διόρθωσης Σφάλματος

Οι κανόνες διόρθωσης σφάλματος προτάθηκαν αρχικά με μοναδικό σκοπό την εκπαίδευση μιας μονάδας μόνο. Αυτοί οι κανόνες ουσιαστικά οδηγούν το output σφάλμα μιας μονάδας στο μηδέν.

#### Perceptron Κανόνας Μάθησης

Ας θεωρήσουμε το perceptron του σχήματος (1.11). Το perceptron αντιστοιχεί ένα input διάνυσμα  $x = [x_1 x_2 \dots x_{n+1}]^T$  σε ένα διπολικό δυαδικό output  $y$  και επομένως μπορεί να θεωρηθεί ως ένας απλός ταξινομητής για δύο κλάσεις. Το input σήμα  $x_{n+1}$  συνήθως ρυθμίζεται σε 1 και παίζει το ρόλο του **bias** στο perceptron. Θα συμβολίζουμε με  $w$  το διάνυσμα  $w = [w_1 w_2 \dots w_{n+1}]^T \in \mathbb{R}^{n+1}$  που αποτελείται από τις παραμέ-

Σχήμα 1.11: The perceptron computational unit

τρος (τα βάρη δηλαδή) του perceptron. Η σχέση μεταξύ input-output του perceptron δίνεται από  $y = \text{sgn}(x^T w)$ , που επιστρέφει  $+1$  ή  $-1$  αναλόγως από το πρόσημο του ορίσματος του διανύσματος, αντίστοιχα.

Υποθέτουμε ότι εκπαιδεύουμε το perceptron αυτό για να φορτώσουμε τα ζεύγη  $\{x^1, d^1\}, \dots, \{x^m, d^m\}$ , όπου  $x^k \in \mathbb{R}^{n+1}$  το  $k$ -οστό input διάνυσμα και  $d^k \in \{-1, +1\}$ ,  $k = 1, 2, \dots, m$  ο επιθυμητός στόχος για το  $k$ -οστό input διάνυσμα (συνήθως η σειρά εκπαίδευσης των διανυσμάτων είναι τυχαία). Αυτά τα διανύσματα αποτελούν το σύνολο εκπαίδευσης (**training set**). Ο στόχος είναι να σχεδιαστεί ένα perceptron ώστε για κάθε input διάνυσμα  $x^k$  του συνόλου εκπαίδευσης, το output  $y^k$  να ταιριάζει με το επιθυμητό  $d^k$ , δηλαδή  $y^k = \text{sgn}(w^T x^k) = d^k$ , για κάθε  $k = 1, 2, \dots, m$ . Σε αυτήν την περίπτωση λέμε ότι το perceptron ταξινομεί σωστά το σύνολο εκπαίδευσης. Φυσικά, σχεδιασμός ενός κατάλληλου perceptron σημαίνει να καθορίσουμε ένα διάνυσμα βαρών  $w^*$  ώστε να ικανοποιούνται οι ακόλουθες συνθήκες:

$$\begin{cases} (x^k)^T w^* > 0, & d^k = +1 \\ (x^k)^T w^* < 0, & d^k = -1 \end{cases} \quad (1.1)$$

Το σύνολο όλων των  $x$  που ικανοποιούν την  $x^T w^* = 0$  ορίζει ένα υπερεπίπεδο στον  $\mathbb{R}^N$ . Επομένως αν βρεθεί ένα διάνυσμα  $w^*$  που να ικανοποιεί την (1.1), αυτό αντιστοιχεί με την εύρεση υπερεπιπέδου που ταξινομεί σωστά όλα τα διανύσματα  $x^k$ ,  $k = 1, 2, \dots, m$ . Με άλλα λόγια, επιθυμούμε ένα υπερεπίπεδο  $x^T w^* = 0$  που διαχωρίζει το χώρο των inputs σε δύο ευκρινείς περιοχές, η μία περιέχει όλα τα σημεία  $x^k$  με  $d^k = +1$

και η άλλη τα  $x^k$  με  $d^k = -1$ .

Μία πιθανή μέθοδος για εύρεση της λύσης  $w^*$  είναι να υλοποιήσουμε τον κανόνα μάθησης του perceptron (Rosenblatt, 1962):

$$\begin{cases} w^1 & \text{αυθαίρετο} \\ w^{k+1} = w^k + \rho(d^k - y^k)x^k & k = 1, 2, \dots \end{cases} \quad (1.2)$$

όπου  $\rho$  είναι μια θετική σταθερά, ο ρυθμός μάθησης. Η διαδικασία της (1.2) έχει ως εξής: αρχικά επιλέγεται ένα διάνυσμα  $w^1$ , τυχαία. Τα  $m$  ζεύγη  $\{x^k, d^k\}$  του συνόλου εκπαίδευσης χρησιμοποιούνται προκειμένου να ενημερώσουν επιτυχώς το διάνυσμα των βαρών, έως ότου μια λύση  $w^*$  βρεθεί ώστε να ταξινομήσει σωστά το σύνολο εκπαίδευσης. Όταν αυτή η ακολουθιακή διαδικασία παρουσίασης των προτύπων εκπαίδευσης γίνει πλήρης, μετά από  $m$  επαναλήψεις δηλαδή, αναφέρεται ως κύκλος (ή πέρασμα) στο σύνολο εκπαίδευσης. Γενικά απαιτούνται περισσότεροι από ένας κύκλοι για να καθοριστεί το κατάλληλο διάνυσμα λύσης. Επομένως, στην εξίσωση (1.2) ο δείκτης  $k$  στο  $w^k$  δηλώνει τον αριθμό των επαναλήψεων, ενώ στο  $x^k$  (και  $d^k$ ) δηλώνει το σετ των προτύπων που εισάγεται στην  $k$  επανάληψη. Για να είμαστε πιο ακριβείς, αν ο αριθμός  $m$  είναι πεπερασμένος, τότε οι δείκτες των  $x^k$  και  $d^k$  πρέπει να αντικατασταθούν με  $[(k-1) \bmod m] + 1$ .

Σημειώνουμε ότι για  $\rho = 0.5$  ο κανόνας του perceptron μπορεί να γραφτεί ως εξής

$$\begin{cases} w^1 & \text{αυθαίρετο} \\ w^{k+1} = w^k + z^k & (z^k)^T w^k \leq 0 \\ w^{k+1} = w^k & \text{διαφορετικά} \end{cases} \quad (1.3)$$

όπου

$$z^k = \begin{cases} +x^k, & d^k = +1 \\ -x^k, & d^k = -1 \end{cases} \quad (1.4)$$

Δηλαδή διόρθωση έχουμε μόνο και μόνο αν συμβεί μια λανθασμένη ταξινόμηση που υποδεικνύεται από

$$(z^k)^T w^k \leq 0 \quad (1.5)$$

Η πρόσθεση του διανύσματος  $z^k$  στο  $w^k$  στη σχέση (1.4) μετακινεί το διάνυσμα των βαρών προς και διαγώνια στο υπερπίπεδο  $(z^k)^T w^k = 0$ .

Σχήμα 1.12: Γεωμετρική αναπαράσταση του κανόνα του perceptron με  $\rho = 0.5$

Το νέο εσωτερικό γινόμενο  $(z^k)^T w^{k+1}$  είναι μεγαλύτερο από το  $(z^k)^T w^k$  κατά το  $\|z^k\|^2$ , και η διόρθωση  $\Delta w^k = w^{k+1} - w^k$  δείχνει μετακίνηση του  $w^k$  στη σωστή κατεύθυνση, όπως φαίνεται στο σχήμα (1.12) <sup>1</sup>.

Επομένως ο κανόνας του perceptron προσπαθεί να βρει μια λύση  $w^*$  για το ακόλουθο σύστημα ανισώσεων:

$$(z^k)^T w > 0 \text{ για } k = 1, 2, \dots, m \quad (1.6)$$

Στην ανάλυση κάθε αλγορίθμου, δύο είναι κυρίως τα θέματα που μας απασχολούν: η ύπαρξη λύσεων και η σύγκλιση του αλγορίθμου στις επιθυμητές λύσεις (εάν υπάρχουν). Στην περίπτωση του perceptron, είναι ξεκάθαρο ότι υπάρχει διάνυσμα λύσης αν και μόνο αν το σύνολο εκπαίδευσης είναι γραμμικώς διαχωρίσιμο. Αν υποθέσουμε πως είναι, μπορούμε να αποδείξουμε ότι ο κανόνας συγκλίνει σε μια λύση (Novicoff 1962, Ridgway 1962, Nilsson 1965):

έστω  $w^*$  η λύση έτσι ώστε

$$(z^k)^T w^* > 0 \text{ για } k = 1, 2, \dots, m \quad (1.7)$$

Τότε από την εξίσωση (1.3), αν το  $k$ -οστό πρότυπο είναι λάθος ταξινομημένο, μπορούμε να γράψουμε

---

<sup>1</sup>Η ποσότητα  $\|z^2\| = z^T z$  δηλώνεται συχνά ως η ενέργεια του  $z$ , όπου  $\|z\|$  η Ευκλείδεια νόρμα του  $z$

$$w^{k+1} - \alpha w^* = w^k - \alpha w^* + z^k \quad (1.8)$$

όπου  $\alpha$  ένας θετικός σταθερός όρος και επομένως

$$\|w^{k+1} - \alpha w^*\|^2 = \|w^k - \alpha w^*\|^2 + 2(z^k)^T(w^k - \alpha w^*) + \|z^k\|^2 \quad (1.9)$$

Αφού το  $z^k$  είναι ταξινομημένο λανθασμένα, έχουμε  $(z^k)^T w^k \leq 0$  και άρα

$$\|w^{k+1} - \alpha w^*\|^2 \leq \|w^k - \alpha w^*\|^2 - 2\alpha(z^k)^T w^* + \|z^k\|^2(w^k - \alpha w^*) + \|z^k\|^2 \quad (1.10)$$

Τώρα αν  $\beta = \max_i \|z^i\|^2$  και  $\gamma = \min_i \|z^i\|^T w^*$  ( $\gamma$  θετικό γιατί  $(z^i)^T w^* > 0$ ) και αντικαταστήσουμε στην (1.10):

$$\|w^{k+1} - \alpha w^*\|^2 \leq \|w^k - \alpha w^*\|^2 - 2\alpha\gamma + \beta^2 \quad (1.11)$$

Αν επιλέξουμε επαρκώς μεγάλο  $\alpha$ , συγκεκριμένα  $\beta^2/\gamma$ :

$$\|w^{k+1} - \alpha w^*\|^2 \leq \|w^k - \alpha w^*\|^2 - \beta^2 \quad (1.12)$$

Επιπλέον η διαφορά των τετραγώνων μεταξύ  $w^k$  και  $\alpha w^*$  μειώνεται τουλάχιστον κατά  $\beta^2$  σε κάθε επανάληψη, και μετά από  $k$  διορθώσεις, ξαναγράφουμε την (1.12):

$$0 \leq \|w^{k+1} - \alpha w^*\|^2 \leq \|w^1 - \alpha w^*\|^2 - k\beta^2 \quad (1.13)$$

Συνεπάγεται ότι η ακολουθία των διορθώσεων πρέπει να τερματιστεί μετά από όχι περισσότερες από  $k_0$  διορθώσεις, όπου

$$k_0 = \frac{\|w^1 - \alpha w^*\|^2}{\beta^2} \quad (1.14)$$

Επομένως, αν υπάρχει λύση, επιτυγχάνεται σε πεπερασμένο αριθμό επαναλήψεων. Γενικά, ένα γραμμικά διαχωρίσιμο πρόβλημα επιτρέπει μη πεπερασμένο αριθμό λύσεων. Ο κανόνας του perceptron της (1.2) συγκλίνει σε μία από αυτές τις λύσεις. Αυτή η λύση όμως είναι ευαίσθητη στις τιμές του  $\rho$  και στη σειρά που εμφανίζονται τα ζευγάρια των input προτύπων.



Ο περιορισμός των διορθώσεων  $k_0$  που δόθηκε στην (1.14) εξαρτάται από την επιλογή του αρχικού διανύσματος των βαρών,  $w^1$ . Αν  $w^1 = 0$ , έχουμε

$$k_0 = \frac{\alpha^2 \|w^*\|^2}{\beta^2} = \frac{\beta^2 \|w^*\|^2}{\gamma^2} \quad (1.15)$$

ή

$$k_0 = \frac{\max_i \|x_i\|^2 \|w^*\|^2}{[\min_i (x^i)^T w^*]^2} \quad (1.16)$$

Όμως η παραπάνω εξίσωση δεν μας προσφέρουν ιδιαίτερη βοήθεια για την πρόβλεψη του μέγιστου αριθμού επαναλήψεων. Ο παρονομαστής της (1.16) όμως αποδεικνύει ότι η δυσκολία του προβλήματος έγκειται στα δείγματα που είναι σχεδόν ορθογώνια στο διάνυσμα της λύσης.

## Η αντικειμενική συνάρτηση του Perceptron

Παρουσιάζει ενδιαφέρον να δούμε πως οι προηγούμενοι κανόνες διόρθωσης μπορούν να παραχθούν από μια μέθοδο απότομης κλίσης σε μια κατάλληλη αντικειμενική συνάρτηση. Για το perceptron, μπορούμε να ορίσουμε την ακόλουθη συνάρτηση (Duda και Hart, 1973):

$$J(w) = - \sum_{z \in Z(w)} z^T w \quad (1.17)$$

όπου  $Z(w)$  το σύνολο των λανθασμένων δειγμάτων του  $w$  (πχ,  $z^T w$ ). Σημειώνουμε ότι αν το  $Z(w)$  είναι κενό, τότε  $J(w) = 0$  (διαφορετικά  $J(w) > 0$ ). Γεωμετρικά, το  $J(w)$  είναι αναλογικό με το άθροισμα των αποστάσεων από τα λάθος ταξινομημένα δείγματα από το σύνορο απόφασης. Όσο πιο μικρό είναι το  $J$ , τόσο καλύτερο θα είναι το διάνυσμα των βαρών  $w$ .

Δοσμένης της συνάρτησης  $J(w)$ , το σημείο αναζήτησης  $w^k$  μπορεί να βελτιωθεί σε κάθε επανάληψη με τη μετατόπισή του κατηφορικά στην επιφάνεια που ορίζει η  $J(w)$  στο χώρο. Μπορούμε δηλαδή να χρησιμοποιήσουμε την  $J$  για να επιλέξουμε μια μέθοδο κλίσης που αναβαθμίζει

το  $w^k$  έτσι ώστε το βήμα επιλέγεται κατηφορικά προς την "πιο απότομη" κατεύθυνση πάνω στην επιφάνεια αναζήτησης  $J(w)$  του  $w^k$ . Αυτό μπορεί να επιτευχθεί κάνοντας το  $\Delta w^k$  ανάλογο με την κλίση της  $J$  στο  $w^k$ , τυπικά γράφουμε <sup>2</sup>

$$w^{k+1} = w^k - \rho \nabla J(w)|_{w=w^k} = w^k - \rho \left[ \frac{\partial J}{\partial w_1} \frac{\partial J}{\partial w_2} \cdots \frac{\partial J}{\partial w_{n+1}} \right]^T \Big|_{w=w^k} \quad (1.18)$$

Εδώ, το αρχικό σημείο  $w^1$  και ο ρυθμός μάθησης  $\rho$  καθορίζονται από το χρήστη. Η εξίσωση (1.18) αποτελεί τη μέθοδο της απότομης κλίσης (**steepest gradient search rule**). Αν αντικαταστήσουμε την κλίση

$$\nabla J(w^k) = - \sum_{z \in Z(w^k)} z \quad (1.19)$$

στην εξίσωση (1.18) προκύπτει ο κανόνας αναβάθμισης των βαρών

$$w^{k+1} = w^k + \rho \sum_{z \in Z(w^k)} z \quad (1.20)$$

Ο αρχικός κανόνας μάθησης της (1.3) μπορεί να θεωρηθεί σαν μια μέθοδος κλίσης για ελαχιστοποίηση της αντικειμενικής συνάρτησης (1.17).

Πρέπει να αναφερθεί ότι από μαθηματική πλευρά, η εξίσωση (1.19) δεν είναι ακριβής. Χάρη στη μορφή της  $J$  (με ξεχωριστά βήματα), απότομες αλλαγές στην κλίση της συμβαίνουν κάθε φορά που το output του perceptron  $y$  μεταβάλλεται. Επομένως, η κλίση της  $J$  δεν ορίζεται στα σημεία "μετάβασης"  $w$  που ικανοποιούν την  $(z^k)^T w = 0$ ,  $k = 1, 2, \dots, m$ . Όμως, λόγω της διακριτής φύσης της (1.18), η πιθανότητα του  $w^k$  να επικαλύψει άλλα σημεία της μετάβασης, είναι αμελητέα και επομένως μπορούμε να εκφράζουμε το  $\nabla J$  όπως στην εξίσωση (1.18).

<sup>2</sup>Οι μέθοδοι της απότομης κλίσης γενικά περιγράφονται από την ακόλουθη εξίσωση:

$$w^{k+1} = w^k - \alpha A \nabla J|_{w=w^k},$$

όπου ο  $A$  είναι ένας  $n \times n$  πίνακας και το  $\alpha$  ένας πραγματικός αριθμός. Η ταχύτητα της σύγκλισης εξαρτάται από την εκλογή του  $\alpha$ . Αναφέρουμε ότι για  $A$  ίσο με τον αντίστροφο της εσσιανής  $[\nabla \nabla J]^{-1}$  και  $\alpha = 1$ , έχουμε την γνωστή μέθοδο του Newton

Σχήμα 1.13: Adaptive Linear Combiner Element (ADALINE)

### Κανόνας Μάθησης Widrow-Hoff ( $\alpha$ -LMS)

Άλλο ένα παράδειγμα κανόνα διόρθωσης σφάλματος μέσω μιας τετραγωνικής συνάρτησης αποτελεί ο κανόνας των Widrow-Hoff (Widrow and Hoff, 1960). Ο κανόνας χρησιμοποιήθηκε αρχικά για εκπαίδευση του ADALINE, σχήμα (1.13)

Σε αυτήν την περίπτωση, το output σε σχέση με το input  $x^k$  είναι απλώς  $y^k = (x^k)^T w$ . Ο κανόνας των Widrow-Hoff αρχικά προτάθηκε σαν ένας κανόνας που ενσωματώνει τη λεγόμενη αρχή της ελάχιστης διαταραχής. Αργότερα, ανακαλύφθηκε (Widrow and Stearns, 1985) ότι αυτός ο κανόνας συγκλίνει στον τετραγωνικό μέσο της λύσης  $w^*$  και αντιστοιχεί στο ελάχιστο άθροισμα τετραγώνων (Least Mean Square - LMS) του output σφάλματος αν όλα τα input πρότυπα έχουν το ίδιο μήκος. Επομένως σε αυτόν τον κανόνα συχνά αποδίδεται το όνομα  $\alpha$ -LMS κανόνας και δίνεται από την

$$\begin{cases} w^1 &= 0 \text{ ή αυθαίρετο} \\ w^{k+1} &= w^k + \alpha(d^k - y^k) \frac{x^k}{\|x^k\|^2} \end{cases} \quad (1.21)$$

όπου  $d^k \in \mathbb{R}$  είναι η επιθυμητή απόκριση, και  $\alpha > 0$ . Η εξίσωση (1.21) είναι παρόμοια με τον κανόνα του perceptron αν θέσουμε  $\rho$  στην εξίσωση (1.2),

$$\rho = \rho^k = \frac{\alpha}{\|x^k\|^2} \quad (1.22)$$

Η σταθερά  $\alpha$  ελέγχει την ευστάθεια και την ταχύτητα της σύγκλισης. Αν τα input διανύσματα είναι ανεξάρτητα στο χρόνο, εγγυάται ευστάθεια για τις περισσότερες περιπτώσεις αν  $0 < \alpha < 2$ .

### Άλλοι Κανόνες βασισμένοι στη μέθοδο της απότομης κλίσης

Ο παρακάτω κανόνας προκύπτει ορίζοντας αρχικά μια κατάλληλη αντικειμενική συνάρτηση και βελτιστοποιώντας την στη συνέχεια με μια επαναληπτική διαδικασία.

### $\mu$ -LMS Κανόνας Μάθησης

Ο  $\mu$ -LMS κανόνας (Widrow και Hoff, 1960) αποτελεί τον πιο αναλυτικό και εφαρμοσμένο απλό κανόνα μάθησης. Έχει εξάλλου ιδιαίτερη σημασία λόγω της πιθανής επέκτασής του στη μάθηση νευρωνικών δικτύων με πολλές μονάδες. Θεωρούμε το σχήμα (1.13). Έστω

$$J(w) = \frac{1}{2} \sum_{i=1}^m (d^i - y^i)^2 \quad (1.23)$$

το μέσο τετραγωνικό σφάλμα της αντικειμενικής συνάρτησης, τότε

$$y^i = (x^i)^T w \quad (1.24)$$

Χρησιμοποιώντας μέθοδο της απότομης κλίσης για να ελαχιστοποιήσουμε την  $J(w)$ ,

$$\begin{aligned} w^{k+1} &= w^k - \mu \nabla J(w) \\ &= w^k + \mu \sum_{i=1}^m (d^i - y^i) x^i \end{aligned} \quad (1.25)$$

Η  $J(w)$  είναι τετραγωνική ως προς τα βάρη εξαιτίας της γραμμικής σχέσης ανάμεσα στα  $y^i$  και  $w$ . Στην πραγματικότητα, η  $J(w)$  ορίζει ένα

κυρτό<sup>3</sup> υπεπαραβαλοειδές με ένα μόνο ελάχιστο  $w^*$  (ολικό). Επομένως, αν η θετική σταθερά  $\mu$  επιλεγεί επαρκώς μικρή, η (1.25) θα συγκλίνει ασυμπτωτικά στη λύση  $w^*$ , ανεξάρτητα από την επιλογή του αρχικού σημείου. Ο κανόνας της (1.25) αναφέρεται συχνά σαν *batch LMS* κανόνας.

Μια εκδοχή της (1.25) είναι και ο  $\mu$ -LMS ή LMS κανόνας:

$$\begin{cases} w^1 &= 0 \text{ ή αυθαίρετο} \\ w^{k+1} &= w^k + \mu(d^k - y^k)x^k \end{cases} \quad (1.26)$$

Παρατηρούμε ότι ο κανόνας είναι ταυτόσημος με τον  $\alpha$ -LMS κανόνα θέτοντας

$$\mu = \mu^k = \frac{\alpha}{\|x^k\|^2} \quad (1.27)$$

Για inputs ανεξάρτητα στο χρόνο η σύγκλιση του μέσου του διάνυσματος βαρών  $\langle w^k \rangle$  εξασφαλίζεται αν ο σταθερός ρυθμός μάθησης  $\mu$  είναι μικρότερος από  $2 / \langle \|x\|^2 \rangle$  (Widrow and Stearns, 1985). Με  $\langle \cdot \rangle$  συμβολίζουμε τη "μέση" ή την αναμενόμενη τιμή. Σε αυτήν την περίπτωση, το  $\langle w^k \rangle$  προσεγγίζει τη λύση  $w^*$  όσο  $k \rightarrow \infty$ .

Σε πρακτικά προβλήματα,  $m > n + 1$ , άρα είναι αδύνατο να ικανοποιηθούν όλες οι απαιτήσεις  $(x^k)^T w = d^k$ ,  $k = 1, 2, \dots, m$ . Επομένως η εξίσωση (1.26) δε συγκλίνει ποτέ. Όμως, για να έχουμε σύγκλιση, θέτουμε  $\mu = \mu_0/k > 0$ , όπου  $\mu_0$  μια μικρή θετική σταθερά.

Όταν ο ρυθμός μάθησης είναι αρκετά μικρός, ο  $\mu$ -LMS κανόνας αποτελεί μια "καλή" προσέγγιση του κανόνα (1.25). Αυτό σημαίνει ότι το διάνυσμα των βαρών  $w^k$  θα τείνει προς το ολικό ελάχιστο  $w^*$  της κυρτής SSE<sup>4</sup> συνάρτησης. Στη συνέχεια, δείχνουμε ότι

$$w^* = X^\dagger d \quad (1.28)$$

όπου  $X = [x^1 \ x^2 \ \dots \ x^m]$ ,  $d = [d^1 \ d^2 \ \dots \ d^m]^T$ , και  $X^\dagger = (XX^T)^{-1}X$  είναι ο γενικευμένος αντίστροφος ή ψευδοαντίστροφος του  $X$  για  $m > n + 1$ .

<sup>3</sup>Μια συνάρτηση της μορφής  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$  λέγεται *κυρτή* αν ικανοποιείται η ακόλουθη συνθήκη:

$$(1 - \lambda)f(u) + \lambda f(v) \geq [(1 - \lambda)u + \lambda v]$$

για κάθε ζεύγος διανυσμάτων  $u$  και  $v$  στον  $\mathfrak{R}^n$  και κάθε πραγματικό  $\lambda$  στο κλειστό διάστημα  $[0,1]$

<sup>4</sup>Sum of Squared Error

Τα ακρότατα (ελάχιστο και μέγιστο) της συνάρτησης  $J(w)$  είναι λύσεις της εξίσωσης

$$\nabla J(w) = 0 \quad (1.29)$$

Άρα κάθε ελάχιστο της SSE συνάρτησης της (1.23) πρέπει να ικανοποιεί την

$$\nabla J(w) = -\sum_{i=1}^m [d^i - (x^i)^T w] x^i = X(X^T w - d) = 0 \quad (1.30)$$

η οποία μπορεί να ξαναγραφτεί ως εξής

$$XX^T w = Xd \quad (1.31)$$

που για μη ιδιάζων πίνακα  $XX^T$  δίνει λύση στην (1.28), αναλυτικά

$$w^* = (XX^T)^{-1} Xd \quad (1.32)$$

Υπενθυμίζουμε ότι επειδή  $w^*$  στην (1.31) ικανοποιεί την  $\nabla J(w^*) = 0$ , δεν εγγυάται πως το  $w^*$  αποτελεί τοπικό ελάχιστο της  $J$ . Ελαχιστοποιεί όμως τις επιλογές αυτές στις οποίες τέτοιο  $w^*$  παρουσιάζει (τοπικά) σημείο ελαχίστου, μεγίστου ή σαγματικό σημείο της  $J$ . Για να εξακριβωθεί ότι το  $w^*$  είναι όντως ελάχιστο της  $J(w)$ , πρέπει να υπολογιστεί η δεύτερη παράγωγος της Εσσιανής

$$\nabla \nabla J = \left[ \frac{\partial^2 J}{\partial w_i \partial w_j} \right]$$

της  $J$  στο  $w^*$  και να αποδειχτεί ότι είναι θετικά ορισμένη<sup>5</sup>. Αυτό το αποτέλεσμα είναι επακόλουθο της ισότητας του  $\nabla \nabla J$  με τον θετικά ορισμένο  $XX^T$ . Επομένως το  $w^*$  αποτελεί ελάχιστο της  $J$ <sup>6</sup>.

<sup>5</sup>Ένας  $n \times n$  πραγματικός συμμετρικός πίνακας  $A$  είναι θετικά ορισμένος αν η τετραγωνική μορφή  $x^T A x$  είναι αυστηρά θετική για όλα τα μη μηδενικά διανύσματα  $x$  στον  $\mathbb{R}^n$

<sup>6</sup>Φυσικά, στο ίδιο αποτέλεσμα θα είχαμε οδηγηθεί σημειώνοντας ότι η κυρτή, τετραγωνική μορφή της  $J(w)$  επιτρέπει ένα ακρότατο  $w^*$ , που πρέπει να είναι ολικό ελάχιστο της  $J(w)$

Ο LMS κανόνας μπορεί επίσης να εφαρμοστεί προκειμένου να συνθέσει το διάνυσμα βαρών  $w$  ενός perceptron για επίλυση προβλημάτων ταξινόμησης με δύο κλάσεις. Ξεκινώντας, εκπαιδεύεται ο γραμμικός νευρώνας του σχήματος (1.13) με δοσμένα ζεύγη εκπαίδευσης  $\{x^k, d^k\}$ ,  $k = 1, 2, \dots, m$  χρησιμοποιώντας τον LMS κανόνα. Κατά την εκπαίδευση, ο επιθυμητός στόχος  $d^k$  ρυθμίζεται σε  $+1$  για τη μια κλάση και σε  $-1$  για την άλλη (στην πραγματικότητα, κάθε θετική σταθερά μπορεί να χρησιμοποιηθεί ως στόχος για τη μια κλάση και κάθε αρνητική για την άλλη). Μετά τη σύγκλιση της διαδικασίας μάθησης, το διάνυσμα της λύσης μπορεί να χρησιμοποιηθεί στο perceptron για ταξινόμηση. Λόγω της μη γραμμικότητας του κατωφλιού στο perceptron, το output του ταξινομητή θα περιοριστεί στο σύνολο  $\{-1, +1\}$ .

Κατά τη χρησιμοποίησή του σαν διάνυσμα βαρών του perceptron, το ελάχιστο της (1.32) δεν ελαχιστοποιεί γενικά το ρυθμό σφάλματος της ταξινόμησης του perceptron. Δεν πρέπει να προκαλεί έκπληξη, αφότου η SSE συνάρτηση δεν είναι σχεδιασμένη να περιορίσει το ελάχιστο μέσα στη γραμμικώς διαχωρίσιμη περιοχή της λύσης. Επομένως δεν αποτελεί απαραίτητα μια γραμμικά διαχωρίσιμη λύση, ακόμα και αν το σύνολο εκπαίδευσης είναι γραμμικά διαχωρίσιμο. Ακόμα όμως και αν το σύνολο εκπαίδευσης δεν είναι γραμμικά διαχωρίσιμο, θα πρόκειται για πολύ χρήσιμη προσέγγιση. Με την εφαρμογή δηλαδή τον LMS στην εκπαίδευση του perceptron η γραμμική διαχωρισιμότητα θυσιάζεται για μια καλή και συμβιβαστική απόδοση σε διαχωρίσιμα και μη διαχωρίσιμα προβλήματα.

## 1.6.2 Μάθηση χωρίς επίβλεψη

Στη μάθηση χωρίς επίβλεψη δεν υπάρχει σήμα οδηγός. Μας δίνεται ένα σύνολο εκπαίδευσης  $\{x^i, i = 1, 2, \dots, m\}$  από διανύσματα του  $\mathbb{R}^n$ . Στόχος είναι να κατηγοριοποιήσουμε ή να ανακαλύψουμε χαρακτηριστικά στα δεδομένα της εκπαίδευσης. Σε κάποιες περιπτώσεις, τα input διανύσματα  $x^i$  πρέπει να αντιστοιχιστούν σε ένα σύνολο προτύπων χαμηλότερης διάστασης, έτσι ώστε να διατηρηθούν στο νέο σύνολο όλες οι τοπολογικές σχέσεις ανάμεσα στα  $x^i$ . Ενδιαφέρον παρουσιάζει η εκπαίδευση δικτύων με απλές μονάδες. Ακολουθούν κάποιοι βασικοί κανόνες χωρίς επίβλεψη για μια μονάδα και για απλά δίκτυα και μελετούνται τρεις κλάσεις κανόνων: **Hebbian learning, competitive learning and self-organizing feature map learning**.

## Hebbian Learning

Οι κανόνες που θα ακολουθήσουν είναι απόρροια της κλασικής υπόθεσης του Hebb (1949). Ο Hebb υπέθεσε ότι οι βιολογικές συναπτικές δυνάμεις ( $\mathbf{w}$ ) μεταβάλλονται σε αναλογία με τη συσχέτιση ανάμεσα στα προ- και μετασυναπτικά σήματα  $\mathbf{x}$  και  $\mathbf{y}$ , αντίστοιχα που εκφράζονται ως (Stent, 1973; Changeux - Danchin, 1976)

$$w^{k+1} = w^k + \rho y^k x^k \quad (1.33)$$

όπου  $\rho > 0$ ,  $x^k$  το input και  $y^k = (x^k)^T w^k$  το output.

Ας υποθέσουμε τώρα πως τα input διανύσματα επιλέγονται τυχαία από μια αυθαίρετη κατανομή πιθανότητας  $p(x)$ . Θεωρούμε επίσης ότι το δίκτυο αποτελείται από ένα μόνο νευρώνα. Σε κάθε επανάληψη  $k$ , παρουσιάζεται ένα διάνυσμα  $\mathbf{x}$ , τυχαία επιλεγμένο από την  $p(x)$ . Εφαρμόζουμε τον κανόνα του Hebb για αναβάθμιση του διανύσματος των βαρών  $\mathbf{w}$ . Η αναμενόμενη μεταβολή των βαρών  $\langle \Delta w \rangle$  μπορεί να υπολογιστεί υπολογίζοντας την (1.33) για όλα τα inputs.

$$\langle \Delta w \rangle = \rho \langle yx \rangle = \rho \langle xx^T w \rangle \quad (1.34)$$

ή υποθέτοντας ότι τα  $\mathbf{x}$  και  $\mathbf{w}$  είναι ανεξάρτητα,

$$\langle \Delta w \rangle = \rho \langle xx^T \rangle \langle w \rangle \quad (1.35)$$

Επομένως, σε κατάσταση ισορροπίας  $\langle \Delta w \rangle = 0$ , η εξίσωση (1.29) οδηγεί στην  $Cw = 0$  και επομένως  $w^* = 0$  είναι η μοναδική κατάσταση ισορροπίας. Ο  $C = \langle xx^T \rangle$  είναι γνωστός σαν αυτοσυσχετιζόμενος πίνακας και δίνεται από την

$$C = \begin{bmatrix} \langle x_1^2 \rangle & \langle x_1 x_2 \rangle & \cdots & \langle x_1 x_n \rangle \\ \langle x_2 x_1 \rangle & \langle x_2^2 \rangle & \cdots & \langle x_2 x_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_n x_1 \rangle & \langle x_n x_2 \rangle & \cdots & \langle x_n^2 \rangle \end{bmatrix} \quad (1.36)$$

Τα στοιχεία της κύριας διαγωνίου του  $\mathbf{C}$  είναι αθροίσματα τετραγώνων των inputs. Ο πίνακας  $\mathbf{C}$  είναι Ερμιτιανός (πραγματικός και συμμετρικός). Άρα οι ιδιοτιμές του  $\lambda_i$ ,  $i = 1, 2, \dots, n$  είναι πραγματικές και μη αρνητικές και έχουν κάθετα ιδιοδιανύσματα  $c^{(i)}$  και κάθε  $c^{(i)}$  ικανοποιεί τη σχέση  $Cc^{(i)} = \lambda_i c^{(i)}$ .



Μπορεί να αποδειχτεί ότι η λύση  $w^* = 0$  δεν είναι ευσταθής, άρα η εξίσωση (1.29) οδηγεί το  $w$  με κατεύθυνση παράλληλη με αυτήν του ιδιοδιανύσματος του  $C$  με τη μεγαλύτερη ιδιοτιμή. Ο κανόνας οδηγείται στη μεγιστοποίηση του τετραγώνου της τυπικής απόκλισης του output μιας γραμμικής μονάδας, αν το  $y$  έχει μηδενικό μέσο (αυτό επιτυγχάνεται αν τα inputs είναι ανεξάρτητες μεταβλητές με μηδενικούς μέσους). Ένας τρόπος για αποφυγή της απόκλισης του κανόνα του Hebb (εξίσωση (1.27)), είναι να κανονικοποιήσουμε το  $\|w\|$  σε 1 μετά από κάθε βήμα (von der Malsburg, 1973; Rubner - Tavan, 1989). Αυτό οδηγεί σε αναβάθμιση του κανόνα

$$\begin{cases} w^1, & \text{αυθαίρετο} \\ w^{k+1} = \frac{w^k + \rho y^k x^k}{\|w^k + \rho y^k x^k\|} \end{cases} \quad (1.37)$$

### Ανταγωνιστική Μάθηση

Τα απλά δίκτυα που βασίζονται στον κανόνα του Hebb είναι ως ένα σημείο ανταγωνιστικά, προκειμένου να συλλάβουν κάποιο κύριο συστατικό του συνόλου εκπαίδευσης. Ακολούθως, επεκτείνεται αυτή η ιδέα ανάμεσα στους νευρώνες και η εξειδίκευσή τους σε συγκεκριμένες κλάσεις προβλημάτων όπως η ομαδοποίηση (clustering) δεδομένων ή κβαντικοποίηση διανυσμάτων (vector quantization). Ένα δίκτυο με outputs με δυαδικές τιμές, με ένα μόνο "on" τη φορά, χρησιμοποιείται για να ταξινομήσει ένα input ως προς την κατηγορία που ανήκει. Αυτές οι κατηγορίες θα ανακαλυφθούν από το δίκτυο με βάση τις συσχετίσεις στα input δεδομένα. Το δίκτυο τότε κατατάσσει κάθε ομάδα "παρόμοιων" δεδομένων σε output κλάση.

### Απλή Ανταγωνιστική Μάθηση

Ας θεωρήσουμε ένα σύνολο από αλληλεπιδρώντες νευρώνες. Ακολουθούμε την απλούστερη αρχιτεκτονική, ένα μόνο επίπεδο νευρώνων, καθένας λαμβάνει το ίδιο input  $x \in \mathbb{R}^n$  και παράγει ένα output  $y_i$ . Υποθέτουμε επίσης, ότι μόνο ένας νευρώνας είναι ενεργός κάθε φορά. Αυτός αποκαλείται νικητής και καθορίζεται ως μεγαλύτερο άθροισμα βαρών,

Σχήμα 1.14: Ανταγωνιστικό δίκτυο ενός επιπέδου

$$net_i^k = w_i^T x^k \quad (1.38)$$

Ο νευρώνας  $i$  είναι ο νικητής αν

$$w_i^T x^k \geq w_j^T x^k \quad \text{για κάθε } j \neq i \quad (1.39)$$

που μπορεί να ξαναγραφτεί ως εξής

$$\|w_i - x^k\| \leq \|w_j - x^k\| \quad \text{για κάθε } j \neq i \quad (1.40)$$

αν  $\|w_i\| = 1$  για κάθε  $i = 1, 2, \dots, m$ . Επομένως νικητής είναι ο κόμβος με το διάνυσμα βαρών που είναι (κατά την Ευκλείδεια έννοια) πιο κοντά στο input διάνυσμα. Η διαφορά πλέον είναι ότι κάθε νευρώνας εμποδίζει τους υπόλοιπους και διεγείρεται μόνος του, όπως φαίνεται στο σχήμα (1.14).

Προκειμένου να εξασφαλίσουμε τη μοναδικότητα του νικητή, πρέπει να ορίσουμε συγκεκριμένα βάρη και συναρτήσεις ενεργοποίησης (Grossberg (1976) και Lippmann (1987)). Μια πιθανή επίλογή για τα βάρη είναι

$$u_{ij} = \begin{cases} 1 & i = j \\ -\epsilon & i \neq j \end{cases} \quad (1.41)$$

Σχήμα 1.15: Συνάρτηση Ενεργοποίησης για το δίκτυο του 1.14

όπου  $0 < \epsilon < 1/m$ , και  $m$  ο αριθμός των νευρώνων στο δίκτυο. Μια κατάλληλη συνάρτηση ενεργοποίησης φαίνεται στο σχήμα (1.15), όπου το  $T$  επιλέγεται ώστε τα outputs  $y_i$  δεν **saturate** στο 1 πριν τη σύγκλιση του νικητή. Μόνο ο νικητής θα **saturate** στο 1, ενώ οι υπόλοιποι θα έχουν μηδενικά outputs.

Για ένα δοσμένο input  $x^k$  που επιλέχθηκε από μια τυχαία κατανομή  $p(x)$ , τα βάρη του νικητή αναβαθμίζονται ( τα υπόλοιπα μένουν αμετάβλητα) ως εξής (Grossberg 1969, von der Malsburg 1973, Rumelhart και Zipser 1985)

$$\Delta w_i = \begin{cases} \rho(x_k - w_i) & \text{αν } w_i \text{ το διάνυσμα του νικητή} \\ 0 & \text{διαφορετικά} \end{cases} \quad (1.42)$$

Ένας πιο κατάλληλος κανόνας είναι ο ακόλουθος

$$\Delta w_i = \rho \left( \frac{x^k}{\|x^k\|} - w_i \right) \quad (1.43)$$

Οι προηγούμενοι κανόνες έχουν την τάση να κλίνουν στο διάνυσμα των βαρών του τρέχοντος νικητή νευρώνα προς την κατεύθυνση του τρέχοντος input. Το συσσωρευμένο αποτέλεσμα από την επαναληπτική

Σχήμα 1.16: Απλή Ανταγωνιστική Μάθηση. (a) Αρχικά διανύσματα βαρών, (b) Διαμόρφωση των βαρών μετά από ανταγωνιστική μάθηση.

εφαρμογή των κανόνων μπορεί να περιγραφεί ως εξής: Αν δούμε τα διανύσματα των inputs και των βαρών σαν κουκκίδες διασκορπισμένες στην επιφάνεια μιας υπερσφαίρας (ή ενός κύκλου, όπως στο σχήμα 1.16), η εφαρμογή του ανταγωνιστικού κανόνα μάθησης έχει ως αποτέλεσμα την ευαισθητοποίηση συγκεκριμένων νευρώνων ως προς γειτονικές ομάδες input δεδομένων. Τελικά, κάποιοι νευρώνες (συνήθως οι νικητές) εξελίσσονται έτσι ώστε το διάνυσμα των βαρών τους να δείχνει προς το "κέντρο του συνόλου" της κοντινότερης σημαντικής ομάδας δεδομένων.

Δεν γνωρίζουμε τον ακριβή αριθμό των ομάδων στο σύνολο των δεδομένων. Συνήθως ο αριθμός τους υπερεκτιμάται και συμπεριλαμβάνονται υπερβολικοί νευρώνες στο δίκτυο. Μετά τη σύγκλιση, μερικοί νευρώνες θα είναι περιττοί, με την έννοια του ότι δεν θα εξελιχθούν σημαντικά και δεν θα εντοπίσουν ομάδες δεδομένων. Κάτι τέτοιο βλέπουμε στο σχήμα (1.16), όπου το  $w_1$  δε μεταβάλλεται κατά τον υπολογισμό. Τέτοιοι νευρώνες είναι πιθανό να είναι ακόμα επιθυμητοί, αφού ίσως εντοπίσουν νέες ομάδες αν η κατανομή  $p(x)$  μεταβληθεί στο χρόνο, ή η πιθανότητα ύπαρξής τους μπορεί να μειωθεί με κάποια **heuristics**.

Αφού επιτευχθεί η σύγκλιση, είναι απαραίτητο να "ρυθμιστεί" το προηγούμενο clustering δίκτυο προκειμένου να καθοριστεί ο αριθμός των νευρώνων που αποτελούν τις ομάδες που έχουν μάθει και τις σχέσεις μεταξύ τους. Εδώ, τα βάρη του δικτύου κρατούνται σταθερά και το σύνολο

Σχήμα 1.17: Διδιάστατα δεδομένα χρησιμοποιούνται για εκπαίδευση του απλού ανταγωνιστικού δικτύου του παραδείγματος

εκπαίδευσης χρησιμοποιείται για να εξετάσει το δίκτυο για ευαισθητοποιημένους νευρώνες.

## Παράδειγμα

Ας θεωρήσουμε το σύνολο των διδιάστατων σημείων του σχήματος (1.17). Αυτό το παράδειγμα περιγράφει την τυπική λύση ενός ανταγωνιστικού δικτύου που αποτελείται από τέσσερις νευρώνες, και εφαρμόζει τις σχέσεις (1.33) και (1.36). Δεν έχει γίνει κανονικοποίηση για τα διανύσματα των βαρών ή τα inputs. Κατά την εκπαίδευση, τα input δείγματα επιλέγονται ομοιόμορφα τυχαία από το σύνολο των δεδομένων. Τα τέσσερα διανύσματα των βαρών αρχικοποιούνται σε τυχαίες τιμές κοντά στο κέντρο του σχήματος (1.17). Η εκπαίδευση με  $\rho = 0.01$  εκτελείται για 1000 επαναλήψεις. Οι τροχιές των διανυσμάτων των βαρών σχεδιάζονται στο σχήμα (1.18a), με τα αρχικά βάρη να διακρίνονται σαν αστερίσκοι. Παρατηρούμε ότι ένας από τους νευρώνες δεν εξελίχθηκε. Αυτό συνέβη διότι δεν έγινε ποτέ νικητής. Το  $\rho$  είναι μικρό,

επομένως τα διανύσματα εισχωρούν και παραμένουν σε "μικροσκοπικές" γειτονιές. Οι διακυμάνσεις των βαρών στις αντίστοιχες περιοχές ενισχύονται αν χρησιμοποιηθεί ένα μεγαλύτερο  $\rho$  ( $\rho = 0.05$ ), όπως φαίνεται στο σχήμα (1.18b). Τελικά, το εκπαιδευμένο δίκτυο ρυθμίζεται απονέμοντας μια μοναδική ετικέτα για κάθε νευρώνα που έχει νικήσει σε ένα ή περισσότερα δείγματα. Εδώ, η εξίσωση (1.33) χρησιμοποιείται για να καθορίσει το νικητή. Το σχήμα (1.19) απεικονίζει το αποτέλεσμα αυτής της διαδικασίας. Χρησιμοποιούνται διαφορετικά σύμβολα προκειμένου να βάλουμε ετικέτες σε κάθε νικητή. Σημειώνουμε ότι ακριβώς οι ίδιες ομάδες του (1.19) δημιουργήθηκαν από το δίκτυο με  $\rho = 0.05$ . Η ομάδα στο δεξιό άκρο του σχήματος (x) έχει αρκετό ενδιαφέρον λόγω της δομής της. Θα περιμέναμε αυτή η ομάδα να χωριστεί σε δύο διακριτές ομάδες από το ανταγωνιστικό δίκτυο. Αν προσέξουμε το σχήμα (1.18b), το δίκτυο προσπάθησε να το κάνει, αλλά απέτυχε.

Σχήμα 1.18: Η ανάπτυξη των τροχιών των βαρών για ένα ανταγωνιστικό δίκτυο τεσσάρων νευρώνων, αν εφαρμοστούν οι εξισώσεις (1.33) και (1.36). Με τους αστερίσκους συμβολίζονται οι αρχικές θέσεις τους. (a) Ρυθμός μάθησης ίσος με 0.01. (b) Ρυθμός μάθησης ίσος με 0.05

Σχήμα 1.19: Μια λύση με τρεις ομάδες για τα δεδομένα του σχήματος (1.17). Η λύση που παρουσιάζεται είναι μια τοπική δομή ομαδοποίησης (clustering) δημιουργημένη από το απλό ανταγωνιστικό δίκτυο του παραδείγματος.

### Κβαντικοποίηση Διανυσμάτων

Μια από τις πλέον συνηθισμένες εφαρμογές της ανταγωνιστικής μάθησης είναι η κβαντικοποίηση διανύσματος για συμπίεση δεδομένων (πχ για δεδομένα εικόνας και ομιλίας). Σε αυτήν την προσέγγιση, ένα δοσμένο σύνολο  $x^k$  δεδομένων (διανυσμάτων) κατηγοριοποιείται σε  $m$  "φόρμες" ("templates") έτσι ώστε αργότερα κάποιος ίσως χρησιμοποιήσει μια κωδικοποιημένη εκδοχή της αντίστοιχης φόρμας κάθε input διανύσματος προκειμένου να απεικονίσει το διάνυσμα, αντί να χρησιμοποιήσει το ίδιο. Αυτό οδηγεί σε αποδοτική κβαντικοποίηση (συμπίεση) μνήμης και μετάδοσης.

Πρόκειται για μια τεχνική όπου ο  $n$ -dimensional χώρος χωρίζεται σε ευκρινείς περιοχές, και για κάθε περιοχή καθορίζεται μια "φόρμα" (διάνυσμα ανακατασκευής - Linde, 1980; Gray, 1984). Όταν παρουσιάζεται με ένα νέο διάνυσμα  $x$ , ένας κβαντικοποιητής καθορίζει πρώτα την περιοχή στην οποία βρίσκεται το διάνυσμα. Στη συνέχεια παράγει μια κωδικοποιημένη



Σχήμα 1.20: Οι διαχωρισμοί ενός Voronoi κβαντικοποιητή με τέσσερα διανύσματα αναδόμησης, τα οποία παρουσιάζονται με ●

εκδοχή του διανύσματος αναδόμησης  $w_i$  που αναπαριστά τη συγκεκριμένη περιοχή που περιέχει το  $x$ . Το σύνολο όλων των πιθανών  $w_i$  συνήθως καλείται **codebook** του κβαντικοποιητή. Όταν χρησιμοποιείται το μέτρο της Ευκλείδιας απόστασης για να αποφασιστεί σε ποιά περιοχή ανήκει το διάνυσμα  $x$ , έχουμε έναν Voronoi κβαντικοποιητή. Ο Voronoi κβαντικοποιητής διαιρεί τον input χώρο σε Voronoi κελιά (Gray, 1984) και κάθε κελί αναπαριστάται από κάποιο από τα διανύσματα αναδόμησης  $w_i$ . Το  $i$ -οστό κελί Voronoi περιέχει τα σημεία αυτά του input χώρου που είναι κοντύτερα (υπό την Ευκλείδια έννοια) στο διάνυσμα  $w_i$  από κάθε άλλο διάνυσμα  $w_j$ ,  $j \neq i$ . Στο σχήμα (1.20) βλέπουμε ένα παράδειγμα των διαχωρισμών του input χώρου ενός Voronoi κβαντικοποιητή για τέσσερα διανύσματα αναδόμησης.

Ο κανόνας μάθησης της (1.36) με καθορισμό του νικητή νευρώνα βασισμένο στην Ευκλείδια απόσταση, όπως στην εξίσωση (1.34), μπορεί να χρησιμοποιηθεί για την κατανομή ενός συνόλου  $m$  διανυσμάτων αναδόμησης  $w_i \in \mathbb{R}^n$ ,  $i = 1, 2, \dots, m$  στον input χώρο των  $n$ -διάστατων διανυσμάτων  $x$ . Έστω ότι τα  $x$  είναι κατανεμημένα σύμφωνα με την πυκνότητα πιθανότητας  $p(x)$ . Αρχικά, θέτουμε τις τιμές των  $w_i$  ίσες με τα  $m$  πρώτα τυχαία δείγματα που έχουν παραχθεί από το  $x$ . Επιπλέον δείγματα  $x$  χρησιμοποιούνται στη συνέχεια για την εκπαίδευση. Ο ρυθμός μάθησης  $\rho$  της εξίσωσης (1.36) επιλέγεται να είναι μια γνησίως φθίνουσα συνάρτηση του αριθμού των επαναλήψεων  $k$ . Βασισμένος σε εμπειρικά αποτελέσματα, ο Kohonen (1989) συμπέρανε πως, κατά μέσο όρο, η ασυ-

μπωτική πυκνότητα του διανύσματος  $w_i$  (ο αριθμός δηλαδή των  $w_i$  που πέφτουν σε μια μικρή ποσότητα του  $\mathbb{R}^n$  με κέντρο το  $x$ ) που εξασφαλίζεται από την παραπάνω ανταγωνιστική διαδικασία, παίρνει τη μορφή μιας συνεχούς, γνησίως αύξουσας συνάρτησης της  $p(x)$ .

Ο Kohonen σχεδίασε (1989) εκδοχές της κβαντικοποίησης διανυσμάτων με επίβλεψη (*Learning Vector Quantization* ή *LVQ*) για προσαρμοσμένη ταξινόμηση προτύπων. Εδώ χρησιμοποιείται πληροφορία της κλάσης για το συντονισμό των διανυσμάτων αναδόμησης σε ένα κβαντικοποιητή Voronoi, ώστε να βελτιωθεί η ποιότητα του ταξινομητή των περιοχών. Στα προβλήματα ταξινόμησης προτύπων, πρέπει να περιγραφεί πιο αναλυτικά η πληροφορία για το σύνορο μεταξύ των κλάσεων των προτύπων παρά το εσωτερικό της. Η παραπάνω διαδικασία μπορεί εύκολα να προσαρμοστεί σε αυτό. Θα μπορούσαμε να ξεκινήσουμε με έναν εκπαιδευμένο κβαντικοποιητή Voronoi και να τον ρυθμίσουμε χρησιμοποιώντας ένα σύνολο ετικετοποιημένων input διανυσμάτων. Κάθε δείγμα ρύθμισης προσδιορίζεται στο  $w_i$  που είναι πιο κοντά. Κάθε  $w_i$  στη συνέχεια, παίρνει ετικέτα ανάλογα με την πλειοψηφία των κλάσεων που παρουσιάζονται από αυτά τα δείγματα και έχουν προσδιοριστεί στο  $w_i$ . Στη συνέχεια, ο συντονισμός των επιφανιών απόφασης επιτυγχάνεται ανταμοίβοντας τις σωστές ταξινομήσεις και τιμωρώντας τις λανθασμένες. Έστω ότι το διάνυσμα εκπαίδευσης  $x^k$  ανήκει στην κλάση  $c_j$ . Υποθέτουμε ότι το κοντινότερο διάνυσμα αναδόμησης  $w_i$  στο  $x^k$  φέρει την ετικέτα της κλάσης  $c_i$ . Τότε μόνο το διάνυσμα  $w_i$  μεταβάλλεται σύμφωνα με τον ακόλουθο κανόνα με επίβλεψη (LVQ rule):

$$\Delta w_j = \begin{cases} +\rho^k(x^k - w_i) & c_j = c_i \\ -\rho^k(x^k - w_i) & c_j \neq c_i \end{cases} \quad (1.44)$$

όπου η  $\rho^k$  θεωρείται γνησίως φθίνουσα συνάρτηση του αριθμού των επαναλήψεων  $k$ . Το αρχικό αποτέλεσμα του κανόνα της (1.44) είναι να ελαχιστοποιήσουμε τον αριθμό των λανθασμένων ταξινομήσεων.

Η ταχύτητα σύγκλισης του LVQ μπορεί να βελτιωθεί αν κάθε διάνυσμα  $w_i$  έχει το δικό του προσαρμοσσιμο ρυθμό μάθησης  $\rho_i^k$  (Kohonen, 1990),

$$\rho_i^k = \begin{cases} \frac{\rho_i^{k-1}}{1+\rho_i^{k-1}} & c_j = c_i \\ \frac{\rho_i^{k-1}}{1-\rho_i^{k-1}} & c_j \neq c_i \end{cases} \quad (1.45)$$

Αυτός ο επαναλαμβανόμενος κανόνας προκαλεί το  $\rho_i$  να μειωθεί αν το  $w_i$  ταξινομεί σωστά το  $x^k$ . Σε διαφορετική περίπτωση, το  $\rho_i$  αυξάνεται. Οι εξισώσεις (1.44) και (1.45) ορίζουν την *OLVQ* (*optimized learning rate LVQ*).

### Self-Organizing Feature Maps (SOFM): Ανταγωνιστική Μάθηση που διατηρεί την τοπολογία

Πρόκειται για μια διαδικασία μη επιβλεπόμενης μάθησης όπου ανακαλύπτονται σημαντικά πρότυπα ή χαρακτηριστικά στα input δεδομένα. Στο περιβάλλον των νευρωνικών δικτύων, αυτή η μάθηση αποτελείται από μεταβολή των συναπτικών βαρών ενός δικτύου από τοπικά αλληλεπιδρώντες νευρώνες ως απόκριση στις εισερχόμενες διεγέρσεις και σε συμφωνία με έναν κανόνα μάθησης μέχρι να αναπτυχθεί μια τελική χρήσιμη διαμόρφωση. Τοπική αλληλεπίδραση μεταξύ των νευρώνων, σημαίνει ότι οι μεταβολές στη συμπεριφορά ενός νευρώνα επηρεάζουν (άμεσα) μόνο τη συμπεριφορά των άμεσα γειτονικών του νευρώνων. Το ερώτημα που προκύπτει είναι το πώς αναπτύσσεται μια χρήσιμη διαμόρφωση. Η απάντηση βρίσκεται σε ένα παρατηρημένο φαινόμενο όπου η γενική τάξη (??? **global order**) μπορεί να προκύψει από τοπικές αλληλεπιδράσεις (Turing, 1952). Αυτό το φαινόμενο έχει εφαρμογή στα νευρωνικά δίκτυα (βιολογικά και τεχνητά) όπου **many originally random local interactions between neighbouring units of a network couple and coalesce into states of global order**.

Στη συνέχεια, παρουσιάζεται ένα δίκτυο που παρουσιάζει self-organization χαρακτηριστικά. Προσπαθεί να αντιστοιχήσει ένα σύνολο input διανυσμάτων  $x^k$  του  $\mathbb{R}^n$  σε έναν πίνακα από νευρώνες (συνήθως μιας ή δύο διαστάσεων) έτσι ώστε να διατηρηθεί κάθε τοπολογική σχέση μεταξύ των  $x^k$  και να παρουσιαστεί από το δίκτυο σε σχέση με την κατανομή των νευρώνων στο χώρο. Αν τα  $x^1$  και  $x^2$  είναι "παρόμοια" ή είναι γειτονικά στον  $\mathbb{R}^n$ , και αν  $r_1$  και  $r_2$  οι θέσεις των αντίστοιχων νικητών νευρώνων στο δίκτυο/πίνακα, τότε η Ευκλείδεια απόσταση  $\|r_1 - r_2\|$  αναμένεται να είναι μικρή. Επίσης η απόσταση τείνει στο μηδέν όσο το

$x^1$  τείνει στο  $x^2$ . Η ιδέα είναι να αναπτύξουμε ένα τοπολογικό χάρτη των input διανυσμάτων έτσι ώστε πανομοιότυπα διανύσματα θα προκαλούν δράση σε γειτονικούς νευρώνες.

Ένα παράδειγμα τέτοιας αντιστοιχίας είναι η αντιστοιχία από τον αμφιβληστροειδή χιτώνα του ματιού στον οπτικό φλοιό. Πιστεύεται ότι τέτοιου είδους βιολογικές αντιστοιχίες δεν προγραμματίζονται από πριν εντελώς από τα γονίδια και ότι υπάρχει κάποιο είδος self-organizing μάθησης και συντονίζει τέτοιες αντιστοιχίες κατά την ανάπτυξη. Τα πρώτα μοντέλα προτάθηκαν από τους von der Malsburg (1973) και Willshaw και von der Malsburg (1976) για το πρόβλημα του αμφιβληστροειδούς. Ακολουθεί το μοντέλο του Kohonen (1982) στο οποίο θα αναφερόμαστε ως *self-organizing feature map* (SOFM).

## Kohonen's SOFM

Σκοπός των SOFM του Kohonen είναι να διατηρήσουν την τοπολογία και την κατανομή της πιθανότητας των input δεδομένων (Kohonen 1982a και 1989). Αυτό το μοντέλο περιλαμβάνει μια αρχιτεκτονική που αποτελείται από μια διδιάστατη δομή γραμμικών νευρώνων, με κάθε νευρώνα να λαμβάνει το ίδιο input  $x^k \in \mathbb{R}^n$ . Κάθε νευρώνας χαρακτηρίζεται από ένα  $n$  - διάστατο διάνυσμα βαρών. Το  $w_i$  διάνυσμα (βάρος του  $i$  - οστού νευρώνα) συχνά θεωρείται σαν διάνυσμα που καθορίζει την "πραγματική θέση" του νευρώνα  $i$  στον  $\mathbb{R}^n$ . Αυτό θα επιτρέψει την ερμηνεία των μεταβολών του  $w_i$  ως κινήσεις του  $i$ . Όμως, πρέπει να έχουμε στο νου μας ότι δεν λαμβάνουν χώρα φυσικές κινήσεις των νευρώνων.

Ο κανόνας μάθησης είναι παρόμοιος με αυτόν της σχέσης (1.42) και ορίζεται ως εξής

$$\Delta w = \rho \Phi(r_i, r_{i^*})(x^k - w_i) \text{ για κάθε } i = 1, 2, \dots \quad (1.46)$$

όπου  $i^*$  ο δείκτης του κερδισμένου νευρώνα. Ο κερδισμένος νευρώνας καθορίζεται ανάλογα με την ευκλείδεια απόσταση της (1.40), χωρίς κανονικοποίηση των βαρών. Η σημαντική διαφορά μεταξύ των δύο κανόνων είναι η παρουσία της περιοχής  $\Phi(r_i, r_{i^*})$  στον τελευταίο. Είναι πολύ σημαντική για την πετυχημένη διατήρηση των τοπολογικών ιδιοτή-

των. Είναι συνήθως συμμετρική (δηλ.  $\Phi(r_i, r_{i^*}) = \Phi(r_i - r_{i^*})$ ) με τιμές κοντά στο 1 για τους  $i$  νευρώνες κοντά στον  $i^*$  και η μονοτονία της είναι φθίνουσα με την Ευκλείδεια απόσταση  $\|r_i - r_{i^*}\|$  ανάμεσα στους  $i$  και  $i^*$  νευρώνες στο δίκτυο.

Κατά το ξεκίνημα της μάθησης, η  $\Phi(r_i, r_{i^*})$  ορίζει μια σχετικά μεγάλη περιοχή όπου όλα τα στοιχεία του δικτύου αναβαθμίζονται για κάθε input  $x^k$ . Όσο προχωράει η διαδικασία της μάθησης, η γειτονιά συρρικνώνεται έως ότου τελικά φτάσει στο μηδέν, και τότε αναβαθμίζεται μόνο ο κερδισμένος νευρώνας. Ο ρυθμός μάθησης πρέπει επίσης να είναι φθίνουσα συνάρτηση προκειμένου να επιτευχθεί σύγκλιση. Θα μπορούσε κανείς να πει ότι μια αρχικά μεγάλη γειτονιά επηρεάζει μια γενική αναζήτηση που αργότερα βελτιώνεται σε τοπική καθώς η απόκλιση της  $\Phi(r_i, r_{i^*})$  πλησιάζει στο μηδέν. Μια επιλογή για την  $\Phi(r_i, r_{i^*})$  είναι η ακόλουθη:

$$\Phi(r_i, r_{i^*}) = e^{-\frac{\|r_i - r_{i^*}\|^2}{2\sigma^2}} \quad (1.47)$$

όπου η τυπική απόκλιση  $\sigma^2$  ελέγχει το πλάτος της γειτονιάς. Εδώ, τα  $\sigma$  και  $\rho$  μπορούν να οριστούν σε αναλογία με  $1/k^\alpha$  ( $0 < \alpha < 1$ ) όπου  $k$  παριστάνει το βήμα. Οι Ritter και Schulten (1988) έκαναν την ακόλουθη πρόταση για την αναβάθμιση του  $\rho$ :

$$\rho^k = \rho_0 \left( \frac{\rho_f}{\rho_0} \right)^{k/k_{max}} \quad (1.48)$$

και για το  $\sigma$ :

$$\sigma^k = \sigma_0 \left( \frac{\sigma_f}{\sigma_0} \right)^{k/k_{max}} \quad (1.49)$$

όπου τα  $\rho_0(\sigma_0)$  και  $\rho_f(\sigma_f)$  ελέγχουν τις αρχικές και τελικές τιμές του ρυθμού μάθησης, αντίστοιχα, και  $k_{max}$  είναι ο μέγιστος αριθμός των βημάτων που περιμένουμε.

Δεν υπάρχει θεωρία για σωστή επιλογή αυτών των παραμέτρων. Στην πράξη, καλές τιμές είναι  $0 < \rho_0 \leq 1$  (τυπικά 0.8),  $\rho_f \ll 1$ ,  $\sigma_0 \approx m_d/2$  ( $m_d$  ο αριθμός των νευρώνων κατά τη μεγαλύτερη διαγώνιο του πίνακα) και  $\sigma_f = 0.5$ . Τελικά το  $k_{max}$  ρυθμίζεται σε 2 ή περισσότερες τάξεις μεγέθους (μεγαλύτερο από το συνολικό αριθμό των νευρώνων στο δίκτυο).

Έστω το input διάνυσμα  $x$  μια τυχαία μεταβλητή με συνάρτηση πυκνότητας πιθανότητας  $p(x)$ . Τότε η βάση της (1.46) και άλλων self-

organizing αλγορίθμων αποτελείται από την ακόλουθη διαδικασία:  
(1) εντοπισμός του πιο ταιριαστού νευρώνα για το input διάνυσμα  $x$ , και (2) αύξηση ταιριάσματος σε αυτόν το νευρώνα και στους γειτονικούς του. Ο υπολογισμός που επιτυγχάνεται από μια επαναλαμβανόμενη εφαρμογή αυτής της διαδικασίας είναι εκπληκτικός και συλλαμβάνεται στην ακόλουθη πρόταση, χάρη στον Kohonen: "Τα  $w_i$  διανύσματα τείνουν να ταξινομηθούν με βάση την αμοιβαία ομοιότητά τους, και η ασυμπτωτική τοπική πυκνότητα του  $w_i$ , είναι της μορφής  $g(p(x))$ , όπου η  $g$  είναι μια συνεχής, γνησίως αύξουσα συνάρτηση".

Πίνακας Ι. Περίληψη των Βασικών Κανόνων Μάθησης

Κανόνας Μάθησης	Συνάρτηση Κριτηρίου	Διάνυσμα Μάθησης
Perceptron rule (supervised)	$J(w) = -\sum_{x^T w \leq 0} z^T w$	$\begin{cases} z^k, & \text{αν } (z^k)^T w^k \leq 0; \\ 0, & \text{διαφορετικά.} \end{cases}$
Perceptron rule with variable learning rate and fixed margin (supervised)	$J(w) = -\sum_{x^T w \leq b} (z^T w - b)$	$\begin{cases} z^k, & \text{αν } (z^k)^T w^k \leq b; \\ 0, & \text{διαφορετικά.} \end{cases}$
Widrow-Hoff rule ( $\alpha$ -LMS) (supervised)	$J(w) = \frac{1}{2} \sum_i \frac{[d^i - (x^i)^T w]^2}{\ x^i\ ^2}$	$[d^k - (x^k)^T w^k] x^k$
$\mu$ -LMS (supervised)	$J(w) = \frac{1}{2} \sum_i [d^i - (x^i)^T w]^2$	$[d^k - (x^k)^T w^k] x^k$
Stochastic $\mu$ -LMS rule (supervised)	$J(w) = \frac{1}{2} \langle [d^i - (x^i)^T w]^2 \rangle$	$[d^k - (x^k)^T w^k] x^k$
Hebbian rule (unsupervised)	$J(w) = -\frac{1}{2} \langle y^2 \rangle$	$y^k x^k$
Standard competitive learning rule (unsupervised)	$J(w) = \frac{1}{2} \sum_k \ x^k - w_{i^*}\ ^2 x^k - w_{i^*}$	
Kohonen's feature map (unsupervised)	$J(w) = \frac{1}{2} \sum_{k,i} \Phi(r, r_{i^*}) \ x^k - w_i\ ^2$	$\Phi(r, r_{i^*}) (x^k - w_i)$

Κανόνας Μάθησης	Συνθήκες	Συνάρτηση Ενεργοποίησης
Perceptron rule (supervised)	$\rho > 0$	$f(\text{net}) = \text{sgn}(\text{net})$
Perceptron rule with variable learning rate and fixed margin (supervised)	$b > 0, \rho^k \geq 0, \sum_{k=1}^{\infty} \rho^k = \infty, \frac{\sum_{k=1}^{\infty} (\rho^k)^2}{(\sum_{k=1}^{\infty} \rho^k)^2} = 0$	$f(\text{net}) = \text{sgn}(\text{net})$
Widrow-Hoff rule ( $\alpha$ -LMS) (supervised)	$\rho^k = \frac{\alpha}{\ x^k\ ^2}, 0 < \alpha < 2$	$f(\text{net}) = \text{net}$
$\mu$ -LMS (supervised)	$0 < \rho < \frac{2}{3\langle \ x\ ^2 \rangle}$	$f(\text{net}) = \text{net}$
Stochastic $\mu$ -LMS (supervised)	$\rho^k \geq 0, \sum_{k=1}^{\infty} \rho^k = +\infty, \sum_{k=1}^{\infty} (\rho^k)^2 < \infty$	$f(\text{net}) = \text{net}$
Hebbian rule (unsupervised)	$\rho \geq 0$	$f(\text{net}) = \text{net}$
Standard competitive learning rule (unsupervised)	$\rho^k \geq 0, \sum_{k=1}^{\infty} \rho^k = \infty, \sum_{k=1}^{\infty} (\rho^k)^2 < \infty$	$f(\text{net}) = \text{net}$
Kohonen's feature map (unsupervised)	$\tau \alpha \rho^k, \sigma^k$ αναπτύσσονται σύμφωνα με την $k^{-\alpha}, f(\text{net}) = \text{net}$	

- Παρατήρηση 1.6.1** • *Perceptron rule (Με επίβλεψη)*: Πεπερασμένος χρόνος σύγκλισης αν το training set είναι γραμμικά διαχωρίσιμο. Το  $\|w\|$  παραμένει περιορισμένο για αυθαίρετα training sets.
- *Perceptron rule with variable learning rate and fixed margin (Με επίβλεψη)*: Συγκλίνει στο  $z^T w > b$  αν το training set είναι γραμμικά διαχωρίσιμο. Πεπερασμένη σύγκλιση αν  $\rho^k = \rho$ , όπου  $\rho$  μια πεπερασμένη θετική σταθερά.
  - *Widrow-Hoff ( $\alpha$ -LMS) (Με επίβλεψη)*: Συγκλίνει στον μέσο των τετραγώνων της ελάχιστης SSE ή LMS λύσης αν  $\|x^i\| = \|x^j\|$ , για κάθε  $i, j$
  - $\mu$  - LMS (Με επίβλεψη): Συγκλίνει στον μέσο των τετραγώνων της ελάχιστης SSE ή LMS λύσης
  - *Stochastic  $\mu$  - LMS (Με επίβλεψη)*:  $\langle \cdot \rangle \equiv$  ο τελεστής του μέσου. (Σε κάθε βήμα το  $x^k$  επιλέγεται τυχαία). Συγκλίνει στον μέσο των τετραγώνων της ελάχιστης SSE ή LMS λύσης
  - *Hebbian (Χωρίς Επίβλεψη)*:  $\|w^*\| \rightarrow \infty$ , με το  $w^*$  στην κατεύθυνση του  $c^{(1)}$  (Παρατήρηση 1.6.2).
  - *Standard competitive learning rule (Χωρίς Επίβλεψη)*: Συγκλίνει σε ένα τοπικό ελάχιστο της  $J$  παρουσιάζοντας clustering σχηματισμό
  - *Kohonen's feature map (Χωρίς Επίβλεψη)*: Τα βάρη αναπτύσσονται προς μια λύση που τείνει να διατηρήσει την τοπολογία του χώρου των input. Η πυκνότητα των σημείων αυτής της λύσης είναι της μορφής  $g(p(x))$ , όπου η  $g$  μια συνεχής αύξουσα συνάρτηση και η  $p(x)$  μια σταθερή πυκνότητα πιθανότητας του  $x^k$ .



**Παρατήρηση 1.6.2** Σημειώνουμε ότι:

$$1. z^k = \begin{cases} x^k, & \text{αν } d^k = +1 \\ -x^k, & \text{αν } d^k = -1 \end{cases}$$

2. Η γενική μορφή της εξίσωσης μάθησης είναι  $w^{k+1} = w^k + \rho^k s^k$ , όπου  $\rho^k$  είναι ο ρυθμός μάθησης και  $s^k$  το διάνυσμα μάθησης.

$$3. net = x^T w$$

4.  $c^i$  είναι το  $i$ -οστό ιδιοδιάνυσμα του αυτοσυσχετιζόμενου πίνακα  $C$  με ιδιοτιμή  $\lambda_i$  ( $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$  και  $\lambda_{max} = \lambda_1$ ).



## Κεφάλαιο 2

# Δυναμικά Νευρωνικά Δίκτυα

### 2.1 Γραμμικός Συσχετιστής

Ενας γραμμικός συσχετιστής, παρουσιάστηκε από τον Anderson [5] και βελτιώθηκε από τον ίδιο και τον Kohonen [6, 7, 8, 18, 19], είναι ένα παράδειγμα μοντέλου νευρωνικού δικτύου που μπορεί να εκμεταλλευτεί ως *συσχετιζόμενη* (ή προσεταιριστική) μνήμη (data addressed). Σε αντίθεση με μια κλασσική μνήμη υπολογιστή όπου το κλειδί για την αναζήτηση ενός αντικειμένου είναι η διεύθυνσή του, στην προσεταιριστική μνήμη η ανάκληση της απαιτούμενης πληροφορίας γίνεται με βάση μέρος της γνώσης (συσχέτιση). Για παράδειγμα, σε εφαρμογές βάσεων δεδομένων η πληροφορία για κάποια αντικείμενα σε μια εγγραφή είναι δυνατή όταν βρίσκουμε την αντίστοιχη συμπληρωμένη εγγραφή. Αντίστοιχα, κάποιος μπορεί να θυμηθεί το χρώμα των ματιών ενός φίλου του ή το όνομα του όταν δει μια ασπρόμαυρη φωτογραφία. Βασικά, θα ξεχωρίσουμε δυο ειδών προσεταιριστικές μνήμες: *αυτοσυσχετιζόμενη* (autoassociative) και *ετερο-συσχετιζόμενη* (heteroassociative). Στην αυτο-συσχετιζόμενη μνήμη η πληροφορία συμπληρώνεται ή ακόμα ανακατασκευάζεται, ενώ στην ετερο-συσχετιζόμενη μνήμη το δεδομένο ανακαλείται. Το παράδειγμα που αναφέρθηκε με την ασπρόμαυρη φωτογραφία, αντιστοιχεί στο να ανακατασκευάσουμε την έγχρωμη φωτογραφία (autoassociation) και να θυμηθούμε το όνομα του ατόμου στη φωτογραφία (heteroassociation), αντίστοιχα.

Η αρχιτεκτονική και υπολογιστική δυναμική του γραμμικού συσχετι-

στή είναι σχεδόν όμοια με αυτήν του MADALINE που περιγράφηκε στην παράγραφο 1.3. Η μοναδική τους διαφορά είναι ότι υπολογίζει γραμμικούς συνδυασμούς των inputs και τελικά η εξίσωση του δικτύου  $y(w):\mathbb{R}$  μπορεί να εκφραστεί ως εξής:

$$y_i = \sum_{i=1}^n w_{ij}x_i \quad j = 1, \dots, m. \quad (2.1)$$

Η γεωμετρική του ερμηνεία είναι ότι τα σχετικά υπερεπίπεδα που αντιστοιχούν στους output νευρώνες, περνούν από τη ρίζα.

Η σχέση (2.1) μπορεί να γραφτεί σε μορφή πινάκων. Τα inputs και outputs του δικτύου θεωρούνται ως διανύσματα στήλες  $\mathbf{x} = (x_1, \dots, x_n)$  μεγέθους  $n \times 1$  και  $\mathbf{y} = (y_1, \dots, y_m)$  μεγέθους  $m \times 1$  αντίστοιχα. Ομοια, η διαμόρφωση του δικτύου δίνεται από ένα πίνακα βαρών  $\mathbf{W}_{m \times n}$ , του οποίου οι σειρές  $\mathbf{w}_j = (w_{j1}, \dots, w_{jn})$  αντιστοιχούν στα εισαγόμενα βάρη των (output) νευρώνων  $j$  ( $j = 1, \dots, m$ ):

$$\begin{pmatrix} w_{11} & \dots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \dots & w_{mn} \end{pmatrix}. \quad (2.2)$$

Επομένως η υπολογιστική δυναμική (2.1) του γραμμικού συσχετιστή μπορεί να περιγραφτεί σαν ένα γινόμενο πινάκων:

$$\mathbf{y} = \mathbf{W}\mathbf{x}. \quad (2.3)$$

Σε adaptive μορφή η επιθυμητή συνάρτηση του γραμμικού συσχετιστή δίνεται πάλι από ένα training set:

$$\mathbb{T} = \left\{ (x_k, d_k) \mid x_k = (x_{k1}, \dots, x_{kn}) \in \mathbb{R} \right. \\ \left. d_k = (d_{k1}, \dots, d_{km}) \in \{0, 1\}^m \quad k = 1, \dots, p \right\} \quad (2.4)$$

Ειδικά στην περίπτωση της αυτο-συσχετιζόμενης μνήμης, το επιθυμητό αποτέλεσμα (output) είναι ίσο με το αρχικό πρότυπο (input pattern), δηλαδή  $m = n$  και  $x_k = d_k$  για  $k = 1, \dots, p$ . Στις επόμενες

παραγράφους, ακολουθούν δύο πιθανές δυναμικές ενός γραμμικού συσχετιστή.

### 2.1.1 Προσαρμογή με τον Κανόνα Hebb

Σε αυτήν την παράγραφο η προσαρμοστική δυναμική (adaptive dynamics) του γραμμικού συσχετιστή περιγράφεται γενικά για την περίπτωση της ετερο-συσχετιζόμενης μνήμης (η οποία περιλαμβάνει την αυτο-συσχετιζόμενη ως ειδική περίπτωση). Ένας από τους πιθανούς τρόπους προσαρμογής του γραμμικού συσχετιστή παρουσιάζεται από τον νευροφυσιολογικό κανόνα του Hebb, ο οποίος λέει ότι η αλλαγή του βάρους που συσχετίζεται με μια σύνδεση μεταξύ δύο νευρώνων, είναι ανάλογη με τις σύμφωνες ενέργειές τους, που εκφράζονται από το γινόμενο των καταστάσεών τους. Με αυτόν τον τρόπο, ο Donald Hebb προσπάθησε να εξηγήσει την αύξηση του επίκτητου αντανακλαστικού [12], όταν η σχεδόν ταυτόχρονη ενεργοποίηση (ή αδράνεια) του πρώτου νευρώνα, που αντιστοιχεί σε μια συνθήκη (πρόταση συλλογισμού) και του δεύτερου νευρώνα που απαγορεύει την ανάκλαση, δυναμώνει το συναπτικό σύνδεσμο με κατεύθυνση από τον πρώτο νευρώνα προς το δεύτερο. Ενώ η αντιθετική ενεργοποίηση αυτών των δύο νευρώνων, εξασθενίζει την αντίστοιχη σύνδεση.

Ο κανόνας του Hebb μπορεί τυπικά να συνοψιστεί στην ακόλουθη (adaptive) δυναμική του γραμμικού συσχετιστή. Στην αρχή της προσαρμογής, σε χρόνο 0, όλα τα βάρη στη διαμόρφωση είναι 0, δηλαδή  $w_{ji}^0 = 0$  ( $j = 1, \dots, m, i = 1, \dots, n$ ). Σύμφωνα με τον κανόνα του Hebb, σε διακριτό χρόνο προσαρμογής  $t = 1, \dots, p$  το  $k$ -στη σειρά - πρότυπο ( $k = t$ ) παρουσιάζεται στο δίκτυο και τα βάρη αναβαθμίζονται σύμφωνα με τον κανόνα του Hebb:

$$w_{ji}^t = w_{ji}^{t-1} + d_{kj}x_{ki} \quad (2.5)$$

όπου  $j = 1, \dots, m, i = 1, \dots, n$ . Όταν ολοκληρωθεί η φάση της προσαρμογής, μετά από  $p$  βήματα, όταν όλα τα όλα τα πρότυπα εκπαίδευσης (training patterns) έχουν μαθευτεί, θα έχουμε την τελική διαμόρφωση:

$$w_{ji} = \sum_{k=1}^p d_{kj}x_{ki} \quad (2.6)$$

Η σχέση (2.5) μπορεί επίσης να γραφτεί σε μορφή πινάκων:

$$W^0 = 0, \quad W^k = W^{k-1} + d_k x_k^T, \quad k = 1, \dots, p \quad (2.7)$$

όπου  $W^k$  η διαμόρφωση του δικτύου σε χρόνο προσαρμογής  $t = k$ . Η τελική διαμόρφωση μπορεί να περιγραφεί σαν γινόμενο πινάκων:

$$W = W^p = \sum_{k=1}^p d_k x_k^T = DX^T \quad (2.8)$$

όπου οι στήλες του  $n \times p$  πίνακα  $X$  και του  $m \times p$  πίνακα  $D$  είναι τα inputs  $x_k$  και τα επιθυμητά αποτελέσματα  $d_k$  των προτύπων (2.4), δηλαδή

$$\begin{pmatrix} x_{11} & \dots & x_{p1} \\ \vdots & \ddots & \vdots \\ x_{1n} & \dots & x_{pn} \end{pmatrix}, \quad \begin{pmatrix} d_{11} & \dots & d_{p1} \\ \vdots & \ddots & \vdots \\ d_{1m} & \dots & d_{pm} \end{pmatrix}. \quad (2.9)$$

Ειδικά για την περίπτωση της αυτο - συσχετιζόμενης μνήμης, όπου  $D = X$ , μπορούμε να γράψουμε:

$$W = XX^T \quad (2.10)$$

Θα θεωρήσουμε ότι τα διανύσματα των προτύπων εκπαίδευσης  $\{x_1, \dots, x_p\}$  είναι ορθογώνια (απαραίτητα  $p \leq n$ ). Αυτό σημαίνει ότι  $x_r^T x_s = 0$  για  $r \neq s$  ( $1 \leq r, s \leq p$ ) και είναι κανονικοποιημένα, δηλαδή  $x_r^T x_r = 1$  ( $r = 1, \dots, p$ ). Αυτό μπορεί να ερμηνευτεί ως εξής: τα συγκεκριμένα πρότυπα διαφέρουν ουσιαστικά λόγω της ορθογωνιότητάς τους και είναι συγκρίσιμα χάρη στο μοναδιαίο μήκος τους. Σύμφωνα με αυτήν την προϋπόθεση, ο γραμμικός συσχετιστής έχει την ιδιότητα της αναπαραγωγής, δηλαδή το δίκτυο ανταποκρίνεται στο αντίστοιχο επιθυμητό output  $d_r$  όταν το αντίστοιχο ερέθισμα  $x_r$  ( $1 \leq r \leq p$ ) εμφανίζεται στην είσοδο. Αυτό μπορεί ναδειχτεί αντικαθιστώντας το  $W$  (σχέση 2.8) στην (2.3):

$$y(x_r) = Wx_r = \left( \sum_{k=1}^p d_k x_k^T \right) x_r = \sum_{k=1}^p d_k (x_k^T x_r) = d_r \quad (2.11)$$

Η ιδιότητα της αναπαραγωγής (2.11) μπορεί να θεωρηθεί ως απαραίτητη προϋπόθεση για τις συσχετιζόμενες μνήμες. Επίσης, ο γραμμικός συσχετιστής θα έπρεπε να ανταποκρίνεται στο επιθυμητό output  $d_r$

ακόμα και για ερέθισμα  $x_r + \delta$ , που είναι πολύ κοντά στο πρότυπο  $x_r$  (δηλαδή η νόρμα  $\|\delta\| = \sqrt{\sum_{i=1}^n \delta_i^2} = \delta$  είναι αρκετά μικρή). Το αντίστοιχο σφάλμα μπορεί να εκφραστεί ως η νόρμα μιας διαφοράς μεταξύ του πραγματικού output για το input  $x_r + \delta$  και του επιθυμητού output  $d_r$ :

$$E_r(\delta) = \|y(x_r + \delta) - d_r\| = \|Wx_r + W\delta - d_r\| = \|W\delta\| \quad (2.12)$$

Εαν κανονικοποιήσουμε και τα επιθυμητά outputs  $d_r$ , δηλαδή  $d_r^T d_r = 1$  ( $r = 1, \dots, p$ ) κάτι που ήδη ισχύει για την αυτο - συσχετιζόμενη μνήμη, αφού τα  $x_r = d_r$  έχουν θεωρηθεί ορθογώνια. Τότε για το σφάλμα  $E_r(\delta)$  από την (2.12) μπορεί να βρεθεί ένα άνω φράγμα χρησιμοποιώντας την τριγωνική ανισότητα ( $\|x + y\| \leq \|x\| + \|y\|$ ) και την ανισότητα Cauchy-Schwarz ( $|x^T y| \leq \|x\| \|y\|$ ):

$$E_r(\delta) = \|W\delta\| \leq \sum_{k=1}^p \|d_k x_k^T \delta\| \leq \sum_{k=1}^p \|d_k\| \|x_k^T\| \|\delta\| = p\delta \leq n\delta \quad (2.13)$$

Αρα,  $E_r(\delta) \rightarrow 0$  για  $\delta \rightarrow 0$  και ο γραμμικός συσχετιστής ανταποκρίνεται ακριβώς στο επιθυμητό output για διεγέρσεις πολύ κοντά στο αρχικό input, άρα μπορεί να χρησιμοποιηθεί ως μια συσχετιζόμενη μνήμη.

### 2.1.2 Pseudohebbian Προσαρμογή

Η ιδιότητα της αναπαραγωγής για το γραμμικό συσχετιστή που υιοθετήθηκε σύμφωνα με τον κανόνα του Hebb, απαιτεί τη συμπληρωματική υπόθεση σχετικά με την ορθογωνιότητα των προτύπων. Επομένως προτάθηκε μια μετατροπή του κανόνα αυτού, η λεγόμενη *pseudohebbian* προσαρμογή [26, 20]. Ομως αυτή η προσέγγιση δεν ανταποκρίνεται πλέον στη νευροφυσιολογική πραγματικότητα. Για λόγους απλότητας, αυτή η εναλλακτική δυναμική του γραμμικού συσχετιστή θα περιγραφεί πρώτα για την αυτο - συσχετιζόμενη μνήμη και μετά θα γενικευτεί και για την ετερο - συσχετιζόμενη.

Ας υποθέσουμε ότι τα διανύσματα  $\{x_1, \dots, x_p\}$ <sup>1</sup> είναι γραμμικώς ανεξάρτητα ( $p \leq n$ ), άρα δημιουργούν μια βάση διανυσμάτων  $V_p$  που είναι υπόχωρος του  $R^n$ . Εξαιτίας της ιδιότητας της αναπαραγωγής (reproduction property), μια ορθογώνια βάση  $\{z_1, \dots, z_p\}$  ενός χώρου διανυσμάτων  $V_p$  δημιουργείται από Gram - Schmidt ορθογωνιοποίηση κατά την pseudohebbian προσαρμογή. Άρα η προσαρμογή του γραμμικού συσχετιστή προχωράει ως εξής: αρχικά, σε χρόνο 0, ο πίνακας των βαρών είναι μηδενικός, δηλαδή  $W^{(0)} = 0$ . Σε διακριτό χρόνο  $t = 1, \dots, p$  όπου το  $k$  - οστό πρότυπο ( $k = t$ ) παρουσιάζεται στο δίκτυο και ορίζεται ένα διάνυσμα  $n \times 1$ :

$$z_k = x_k - W^{(k-1)}x_k \quad (2.14)$$

και η αναβάθμιση των βαρών γίνεται ως εξής:

$$W^k = W^{(k-1)} + \frac{z_k z_k^T}{z_k^T z_k} \quad (2.15)$$

Ο πίνακας βαρών μπορεί να εκφραστεί πάλι σαν ένα πεπερασμένο άθροισμα πινάκων:

$$W = W^{(p)} = \sum_{k=1}^p \frac{z_k z_k^T}{z_k^T z_k} = X X^+ \quad (2.16)$$

όπου  $X$  ο πίνακας από την (2.9) και

$$X^+ = (X^T X)^{-1} X^T \quad (2.17)$$

ο ψευδο - αντίστροφός του.

Η γεωμετρική ερμηνεία της pseudohebbian προσαρμογής (2.14), (2.15), στο  $k$  βήμα, απεικονίζεται στο σχήμα (2.1):

Θα αποδείξουμε με μαθηματική επαγωγή στο  $k$ , ότι το διάνυσμα  $W^{(k-1)}x_k$  είναι η ορθογώνια προβολή του  $x_k$  στο διανυσματικό χώρο  $V_{k-1}$  που ορίζεται από τη βάση  $x_1, \dots, x_{k-1}$  ή την ορθογώνια βάση  $z_1, \dots, z_{k-1}$ . Σημειώνουμε ότι το διάνυσμα  $x_k$  δεν ανήκει στη βάση  $V_{k-1}$ , αφού τα διανύσματα  $x_1, \dots, x_p$  είναι γραμμικώς ανεξάρτητα. Άρα θέλουμε να δείξουμε ότι το διάνυσμα  $z_k = x_k - W^{(k-1)}x_k$  είναι κάθετο στα

<sup>1</sup>τα οποία συμπίπτουν με τα επιθυμητά outputs στην περίπτωση της αυτο - συσχετιζόμενης



Σχήμα 2.1: Γεωμετρική Ερμηνεία της pseudohebbian στρατηγικής

διανύσματα της βάσης  $z_r$  ( $r = 1, \dots, k-1$ ). Αυτό σημαίνει ότι θέλουμε να δείξουμε ότι  $z_k^T z_r = 0$  για  $r = 1, \dots, k-1$ . Μετά τη μεταφορά του μέλους  $\sum_{s=1}^{k-1} \frac{z_s z_s^T}{z_s^T z_s}$  από την (2.15) αντικαθιστούμε για  $((W^{(k-1)})^T)$  στην  $z_k^T = x_k^T - x_k^T ((W^{(k-1)})^T)$ . Ακόμη, γνωρίζουμε ότι η  $z_1, \dots, z_{k-1}$  είναι ορθογώνια, λόγω της υπόθεσης, άρα έχουμε:

$$z_k^T z_r = x_k^T z_r - \sum_{s=1}^{k-1} \frac{x_k^T (z_s z_s^T) z_r}{z_s^T z_s} = x_k^T z_r - \frac{x_k^T z_r (z_r z_r^T)}{z_r^T z_r} = 0. \quad (2.18)$$

Προκύπτει από τα παραπάνω ότι το διάνυσμα  $x$  συμπίπτει με την ορθογώνια προβολή του  $Wx = x$  όταν βρίσκεται στο χώρο  $V_p$ . Ειδικά,  $y(x_r) = Wx_r = x_r$ , για διανύσματα βάσης  $\{x_1, \dots, x_p\}$ . Αυτό σημαίνει ότι το νευρωνικό δίκτυο του γραμμικού συσχετιστή που προκύπτει από την pseudohebbian προσαρμογή (2.14), (2.15) έχει την ιδιότητα της αναπαραγωγής. Επίσης το αποτέλεσμα του δικτύου  $y(x_r + \delta) = W(x_r + \delta)$  για το input  $x_r + \delta$  κοντά στο  $r$ -στη σειρά - πρότυπο  $x_r$ , είναι η ορθογώνια προβολή του στο χώρο  $V_p$ . Αφού αντικαταστήσουμε το  $W$  από την (2.16), μπορούμε να βρούμε ένα άνω φράγμα για το σφάλμα  $E_r(\delta) = \|y(x_r + \delta) - x_r\| = \|Wx_r + W\delta - x_r\| = \|W\delta\|$  όπως κάναμε και στην (2.13):

$$E_r(\delta) = \|W\delta\| \leq \sum_{s=1}^{k-1} \frac{\|z_k z_k^T \delta\|}{\|z_k\|^2} \leq p\delta \leq n\delta \quad (2.19)$$

όπου  $\delta = \|\delta\|$ , επομένως,  $E_r(\delta) \rightarrow 0$  για  $\delta \rightarrow 0$ . Τελικά ο τύπος (2.16) γενικεύεται για την περίπτωση της ετερο - συσχετιζόμενης μνήμης:

$$W = DX^+ = D(X^T X)^{-1} X^T \quad (2.20)$$

όπου  $D, X$  οι πίνακες της (2.9). Η αναλογία των αναδρομικών σχέσεων (2.14), (2.15) στην περίπτωση της ετερο - συσχετιζόμενης μνήμης από το θεώρημα του Greville:

$$W^k = W^{(k-1)} + \frac{(d_k - W^{(k-1)} x_k) z_k^T}{z_k^T z_k} \quad (2.21)$$

με το  $z_k$  να είναι το ίδιο διάνυσμα στήλη όπως στην περίπτωση της αυτο - συσχετιζόμενης μνήμης. Χρησιμοποιώντας τον ψευδο - αντίστροφο (σχέση (2.17)), το  $z_k$  μπορεί εναλλακτικά να εκφραστεί ως εξής:

$$z_k = x_k - X^{(k-1)} (X^{(k-1)})^+ x_k \quad (2.22)$$

όπου ο  $X^{(k-1)}$  είναι ένας  $n \times (k-1)$  πίνακας, με στήλες τα input διανύσματα  $x_1, \dots, x_{k-1}$  των  $k-1$  πρώτων προτύπων (2.4):

$$X^{(k-1)} = \begin{pmatrix} x_{11} & \dots & x_{k-1,1} \\ \vdots & \ddots & \vdots \\ x_{1n} & \dots & x_{k-1,n} \end{pmatrix}. \quad (2.23)$$

Εκτός από ετερο - συσχετιζόμενο δίκτυο της (2.21), ο υπολογισμός του  $z_k$  σύμφωνα με την (2.14) απαιτεί ένα επιπλέον μοντέλο αυτο - συσχετιζόμενου δικτύου με πίνακα βαρών αυτόν της σχέσης (2.15). Επομένως η pseudohebbian δυναμική (2.20) για ετερο - συσχετιζόμενη μνήμη (δηλαδή για τον υπολογισμό του  $W = DX^+$  προσεγγίζεται συχνά από τον κανόνα του Widrow.

Τέλος η pseudohebbian δυναμική (2.20) βεβαιώνει την ικανότητα του γραμμικού ετερο - συσχετιστή για αναπαραγωγή των προτύπων εκπαίδευσης (2.4). Για επιβεβαίωση του κανόνα αναπαραγωγής, με  $[X_r]$  δηλώνουμε την  $r$  - στήλη του πίνακα  $X$ :

Σχήμα 2.2: Η τοπολογία ενός δικτύου Hopfield

$$y(x_r) = Wx_r = DX^+[X_r] = D[(X^T X)^{-1} (X^T X)]_r = d_r. \quad (2.24)$$

## 2.2 Δίκτυο Hopfield

Ένα ακόμη σημαντικό μοντέλο αυτο - συσχετιζόμενου νευρωνικού δικτύου, είναι το δίκτυο Hopfield. Είχε ήδη παρουσιαστεί από τους McCulloch και Pitts [22] και αργότερα εξετάστηκε λεπτομερώς από τους Amari [2], W. A. Little και G. L. Shaw [21]. Όμως μόνο χάρη στον John Hopfield [15], που εκμεταλλεύτηκε μια σαφή αναλογία με φυσικές θεωρίες μαγνητικών υλικών για αναλύσει τη σταθερότητα αυτού του δικτύου, έγινε ευρέως γνωστό. Έτσι απέκτησε και το όνομά του. Ένα δίκτυο Hopfield μπορεί να χρησιμοποιηθεί ως μια αυτο - συσχετιζόμενη μνήμη.

### 2.2.1 Βασικό Μοντέλο

Αρχικά η αρχιτεκτονική δυναμική ενός δικτύου Hopfield ορίζει μια πλήρη τοπολογία ενός κυκλικού δικτύου με  $n$  νευρώνες. Κάθε νευρώνας συνδέεται με όλους τους νευρώνες του δικτύου και όλοι οι νευρώνες λειτουργούν ως input και output ταυτόχρονα.

Σημειώνουμε με  $\xi_1, \dots, \xi_n \in Z$  το ακέραιο επίπεδο διέγερσης και με  $y_1, \dots, y_n \in \{-1, 1\}$  τις διπολικές καταστάσεις όλων των νευρώνων. Κάθε σύνδεση από τον νευρώνα  $i$  στον  $j$  ( $i, j = 1, 2, \dots, n$ ) σημειώνεται με ένα ακέραιο συναπτικό βάρος  $w_{ji} \in Z$ . Στο βασικό μοντέλο οι πολώσεις (biases) παραλείπονται, δηλαδή όλα τα κατώφλια είναι 0. Ακόμα, δεν υπάρχουν συνδέσεις από ένα νευρώνα προς τον εαυτό του, δηλαδή  $w_{jj} = 0$  ( $j = 1, \dots, n$ ).

Περιγράφουμε πρώτα τη δυναμική που βασίζεται στον κανόνα του Hebb. Η επιθυμητή συνάρτηση του δικτύου καθορίζεται πάλι από ένα σύνολο προτύπων  $p$ , καθένα από τα οποία δίνεται από ένα διάνυσμα από  $n$  διπολικές καταστάσεις input και output νευρώνων:

$$T = \{x_k = (x_{k1}, \dots, x_{kn}) \in \{-1, 1\}^n, k = 1, \dots, p\} \quad (2.25)$$

Η φάση προσαρμογής του δικτύου, σύμφωνα με τον κανόνα του Hebb, γίνεται σε  $p$  διακριτά βήματα, στα οποία παρουσιάζονται ένα - ένα τα πρότυπα. Η διαμόρφωση μπορεί να περιγραφεί ως εξής (μπορεί να γίνει σύγκριση με την (2.6):

$$w_{ji} = \sum_{k=1}^p x_{kj} x_{ki} \quad 1 \leq j \neq i \leq n \quad (2.26)$$

Σημειώνουμε πως  $w_{ji} = w_{ij}$  ( $1 \leq i, j \leq n$ ) αφού οι θέσεις των νευρώνων  $i, j$  στον τύπο (2.26) είναι συμμετρικές. Επομένως, συχνά το δίκτυο Hopfield καλείται *συμμετρικό νευρωνικό δίκτυο*. Το βάρος  $w_{ji} = w_{ij}$  αναπαριστά τη διαφορά ανάμεσα στον αριθμό των συμφωνούντων καταστάσεων  $x_{kj} = x_{ki}$  (δηλαδή  $x_{kj} x_{ki} = 1$ ) των νευρώνων  $i, j$  στα πρότυπα που ενισχύουν την αντίστοιχη σύνδεση, και ανάμεσα στις καταστάσεις  $x_{kj} \neq x_{ki}$  αντίστοιχα. Το αποτέλεσμα φαίνεται στο πρόσημο του βάρους  $w_{ij}$ , που είναι θετικό αν ο αριθμός των καταστάσεων που συμφωνούν υπερσχύει από τον αριθμό των καταστάσεων που διαφωνούν. Η υπολογιστική δυναμική του δικτύου Hopfield περιγράφεται για την περίπτωση διαδοχικών σύγχρονων υπολογισμών. Σε χρόνο 0, οι νευρώνες παίρνουν το πρόσημο του input του δικτύου  $x = (x_1, \dots, x_n)$ , δηλαδή  $y_i^0 = x_i$  ( $i = 1, \dots, n$ ). Για  $t > 0$  αναβαθμίζεται μόνο ένας νευρώνας  $j$ . Για παράδειγμα αυτός ο νευρώνας επιλέγεται έτσι ώστε  $t = \tau n + j$ , όπου  $\tau$  είναι ο λεγόμενος μακροσκοπικός χρόνος, που μετράει τον αριθμό των

epochs στα οποία αναβαθμίζονται όλοι οι νευρώνες. Αρχικά υπολογίζεται το **integer excitation level** του νευρώνα  $j$ :

$$\xi_j^{(t-1)} = \sum_{i=1}^n w_{ji} y_i^{(t-1)} \quad (2.27)$$

και το πρόσημό του καθορίζει τη νέα διπολική κατάστασή του:

$$y_j^{(t)} = \begin{cases} 1, & \text{if } \xi_j^{(t-1)} > 0 \\ -1, & \text{if } \xi_j^{(t-1)} < 0 \end{cases} \quad (2.28)$$

Ο κανόνας (2.28) μπορεί να τροποποιηθεί έτσι ώστε η κατάσταση  $y_j^{(t)} = y_j^{(t-1)}$  του νευρώνα  $j$  να παραμείνει αμετάλλαχτη για  $\xi_j^{(t-1)} = 0$ , αντί να προτιμήσουμε την κατάσταση  $y_j^{(t)} = 1$  όπως προτείνει η σχέση (2.28). Ο υπολογισμός του δικτύου Hopfield τερματίζεται σε χρόνο  $t^*$  όταν το δίκτυο βρίσκεται σε μια σταθερή κατάσταση (stable state), κάτι που σημαίνει ότι οι καταστάσεις του δεν αλλάζουν άλλο:  $y_j^{(t^*+n)} = y_j^{(t^*)}$ . Τότε οι καταστάσεις του (output) νευρώνα αναπαριστούν το output του δικτύου  $y = (y_1, \dots, y_n)$  όπου  $y_j = y_j^{(t^*)}$  ( $j = 1, \dots, n$ ). Μπορεί να αποδειχθεί, με την προϋπόθεση της συμμετρίας των βαρών, ότι ο σειριακός υπολογισμός ενός Hopfield που ελέγχεται από τις (2.27), (2.28), τερματίζεται για κάθε input. Άρα το δίκτυο Hopfield υπολογίζει μια συνάρτηση  $y(w) : \{-1, 1\}^n \rightarrow \{-1, 1\}^n$  στο χώρο των inputs. Αυτή εξαρτάται από τη διαμόρφωση των βαρών, καθώς και από τη σειρά των αναβαθμίσεων των νευρώνων.

Επίσης μπορούμε να θεωρήσουμε ένα παράλληλο υπολογισμό του Hopfield, σε ένα υπολογιστικό βήμα, οι καταστάσεις των περισσότερων νευρώνων αναβαθμίζονται σύμφωνα με τις (2.27), (2.28). Όμως σε αυτήν την περίπτωση γενικά μπορεί ο υπολογισμός να μην τερματιστεί και το δίκτυο να τροποποιήσει δύο διαφορετικές καταστάσεις μετά από λίγο χρόνο. Ακόμη, στο ασύγχρονο μοντέλο Hopfield οι νευρώνες αναβαθμίζονται ανεξάρτητα σε τυχαία σειρά, σε αντίθεση με τις συστηματικές αναβαθμίσεις (των σειριακών ή παράλληλων μοντέλων).

## 2.2.2 Συνάρτηση Ενέργειας

Όπως ήδη αναφέρθηκε, το δίκτυο Hopfield έχει μια φυσική υλική αναλογία. Μερικά απλά μοντέλα μαγνητικών υλικών στη στατιστική μηχανική

Σχήμα 2.3: Απλοποιημένο μοντέλο μαγνητικού υλικού.

μπορούν να θεωρηθούν ως Hopfield δίκτυα. Ένα μαγνητικό υλικό σε αυτά τα μοντέλα μπορεί να περιγραφεί σαν ένα σετ μαγνητών (οι λεγόμενες περιστροφές - spins) που αντιστοιχούν σε νευρώνες Hopfield. Αυτά αναπαριστούν τη μορφή κρυστάλλου των υλικών, όπως φαίνεται στο σχήμα (2.3).

Στην πιο απλή μορφή ατόμου, κάθε περιστροφή μπορεί να έχει δύο διαφορετικούς μαγνητικούς προσανατολισμούς, οι οποίοι μοντελοποιούνται από το διπολικό νευρώνα σε 1 και -1 στο δίκτυο Hopfield. Το υλικό μοντέλο περιγράφεται περισσότερο από αλληλεπιδράσεις και δυναμικές των περιστροφών. Κάθε περιστροφή επηρεάζεται από ένα μαγνητικό, που μπορεί να χωριστεί σε ένα εξωτερικό πεδίο που αντιστοιχεί στο input του Hopfield, και στο εσωτερικό που παράγεται από τις άλλες περιστροφές. Οι συμβολές του κάθε ατόμου στο περιβάλλον εσωτερικό πεδίο είναι ανάλογες με την περιστροφή του. Άρα η επιρροή του μαγνητικού πεδίου σε μια δοσμένη περιστροφή καθορίζεται από το άθροισμα των αντίστοιχων συνεισφορών (contributions) και αντιστοιχεί στην υπολογιστική δυναμική των τύπων (2.27), (2.28) του Hopfield. Τα συναπτικά βάρη στον τύπο (2.27) μετρούν τη δύναμη των αμοιβαίων αλληλεπιδράσεων ανάμεσα

στις περιστροφές. Αυτοί οι συντελεστές είναι απαραίτητα συμμετρικοί στο υλικό μοντέλο και μπορούν να είναι ευκρινείς, θετικοί ή αρνητικοί, ανάλογα με τις μακροσκοπικές ιδιότητες του υλικού (προφανώς το υλικό μοντέλο αντιστοιχεί στο ασύγχρονο δίκτυο Hopfield).

Για καλύτερη κατανόηση των (2.27), (2.28), ο Hopfield όρισε τη λεγόμενη **συνάρτηση ενέργειας** (energy function)  $\mathbf{E}(\mathbf{y})$ . Αυτή η συνάρτηση συσχετίζει κάθε κατάσταση  $y \in \{-1, 1\}^n$  με την πιθανή ενέργεια βάσει του ακόλουθου δευτεροβάθμιου πολυωνύμου:

$$E(y) = -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ij} y_j y_i \quad (2.29)$$

Από τον ορισμό της συνάρτησης ενέργειας, βλέπουμε πως οι καταστάσεις του δικτύου με χαμηλή ενέργεια (δηλαδή αρκετά μεγάλα  $w_{ji} y_j y_i$ , π.χ. θετικά) έχουν μεγαλύτερη σταθερότητα, αφού το πρόσημο του βάρους  $w_{ji}$  που συσχετίζεται με τη σύνδεση μεταξύ των νευρώνων  $j$  και  $i$  είναι σταθερό με την αμοιβαία σχέση μεταξύ των καταστάσεών τους  $y_j$  και  $y_i$ . Ξέρουμε δηλαδή πως το θετικό βάρος αναγκάζει τους νευρώνες να συμφωνήσουν, ενώ το αρνητικό τους κάνει να διαφωνήσουν. Αντίθετα, οι καταστάσεις με υψηλή ενέργεια είναι ασταθείς για τον ίδιο λόγο. Χάρη σε αυτήν την ιδιότητα, η συνάρτηση ενέργειας με το αντίθετο πρόσημο συχνά καλείται *σταθερότητα ή αρμονία* (stability or harmony).

Για πρακτικούς λόγους θα θεωρήσουμε τη συνάρτηση ενέργειας συνεχή, αν και μόνο διακριτές τιμές θεωρούνται για το παραπάνω βασικό μοντέλο του Hopfield. Αρχικά το δίκτυο εφοδιάζεται με ενέργεια  $E(y^{(0)})$  με τη βοήθεια της αρχικής κατάστασης του δικτύου  $y^{(0)} = x$  (το input του δικτύου). Στην πορεία του υπολογισμού αυτή η ενέργεια καταναλώνεται, δηλαδή η συνάρτηση ενέργειας  $E(y^{(t)}) > E(y^{(t+1)})$  μειώνεται στο χρόνο, έως ότου το δίκτυο φτάσει σε μια σταθερή κατάσταση  $y^{(t^*)}$ , σε χρόνο  $t^*$ , που αντιστοιχεί σε ένα τοπικό ελάχιστο της συνάρτησης ενέργειας  $E(y^{(t^*)})$ . Είναι ενδιαφέρον το γεγονός πως η μείωση της συνάρτησης ενέργειας (2.29) είναι όμοια με την ελαχιστοποίηση του "σφάλματος"  $E(r)$  σε μια μέθοδο κλίσης. Συγκεκριμένα για  $t > 0$ , η νέα κατάσταση  $y_j^{(t)}$  ενός επιλεγμένου νευρώνα  $j$  αντιστοιχεί στο πρόσημο της αρνητικής κλίσης (2.30) στο σημείο  $y_j^{(t-1)}$ :

$$-\frac{\partial E}{\partial y_j}(y_j^{(t-1)}) = \sum_{j=1}^n w_{ji}y_i^{(t-1)} \quad (2.30)$$

Συγκριτικά με τα multilayered νευρωνικά δίκτυα που εκπαιδεύονται με την backpropagation, τα Hopfield έχουν αντίθετο χαρακτήρα. Ενώ το δίκτυο Hopfield υπό τον κανόνα του Hebb (2.26) παρουσιάζει μια μη - επαναλαμβανόμενη διαδικασία της οποίας η διάρκεια εξαρτάται μόνο από τον αριθμό των προτύπων εκπαίδευσης, ο αλγόριθμος backpropagation πραγματοποιεί μια επαναληπτική διαδικασία προκειμένου να ελαχιστοποιήσει το σφάλμα του δικτύου (με τη μέθοδο της κλίσης), χωρίς καμιά εγγύηση για τη σύγκλιση. Από την άλλη μεριά, ο χρόνος που καταναλώνεται για τη φάση του υπολογισμού ενός multilayered δικτύου δίνεται μόνο από τον αριθμό των επιπέδων, ενώ ο τρόπος υπολογισμού ενός δικτύου Hopfield (2.27), (2.28), είναι μια επαναληπτική διαδικασία που ελαχιστοποιεί την ενέργεια του δικτύου, με μια διακριτή παραλλαγή της μεθόδου της κλίσης που μπορεί γενικά να μη συγκλίνει (π.χ. παράλληλος υπολογισμός).

Ο σκοπός της προσαρμογής του δικτύου σύμφωνα με τον κανόνα του Hebb (2.26) είναι να βρούμε μια διαμόρφωση του δικτύου ώστε η αντίστοιχη συνάρτηση του δικτύου βάζει σε εφαρμογή μια αυτο - συσχετιζόμενη μνήμη κατά τη φάση του υπολογισμού. Αρα το δίκτυο Hopfield θα έπρεπε να βγάζει ως αποτέλεσμα ένα πρότυπο για κάθε input που είναι κοντά σε αυτό το πρότυπο. Από την άποψη της ενέργειας, κάθε πρότυπο της (2.25) θα έπρεπε να είναι ένα τοπικό ελάχιστο της συνάρτησης ενέργειας, ένα σταθερό σημείο, δηλαδή, του δικτύου. Όλα τα inputs που είναι κοντά στο αντίστοιχο πρότυπο, τοποθετούνται σε μια γειτονιά αυτής της σταθερής κατάστασης και σχηματίζουν τη λεγόμενη περιοχή έλξης (attraction area). Η περιοχή έλξης μιας σταθερής κατάστασης είναι ένα σύνολο από αρχικές καταστάσεις του δικτύου, που αν το δίκτυο ξεκινήσει με αυτές, φτάνει σε σταθερή κατάσταση, κατά τη φάση του υπολογισμού. Στη γεωμετρική ερμηνεία, η επιφάνεια ενέργειας χωρίζεται σε περιοχές έλξης συγκεκριμένων τοπικών ελαχίστων και η συνάρτηση του δικτύου αντιστοιχεί κάθε input, από μια περιοχή έλξης κάποιου τοπικού ελαχίστου, στο ελάχιστό του. Παριστάνεται στο σχήμα (2.4) όπου σημειώνονται τα τοπικά ελάχιστα των περιοχών έλξης.



Ομως, σαν αποτέλεσμα της μάθησης με τον κανόνα του Hebb, εμφανίζονται αυθόρμητα πρόσθετα τοπικά ελάχιστα, οι λεγόμενες νόθες καταστάσεις (*spurious states*), που δεν αντιστοιχούν σε κάποιο από τα πρότυπα. Υπάρχουν παραλλαγές του Hopfield, στις οποίες οι νόθες καταστάσεις μπορούν να αποφευχθούν από τη μάθηση του δικτύου. Για παράδειγμα, στο δίκτυο με διαμόρφωση  $w_{ji}$  ( $1 \leq j, i \leq n$ ), μια νόθα κατάσταση  $x' = (x'_1, \dots, x'_n) \in \{-1, 1\}^n$  μπορεί να αποφευχθεί, με την τροποποίηση του κανόνα του Hebb [33], που παράγει νέα βάρη  $w'_{ji}$  ως εξής:

$$w'_{ji} = w_{ji} - x'_j x'_i \quad 1 \leq j \neq i \leq n \quad (2.31)$$

Στη νευροφυσιολογία, απαλλαγή των νευρωνικών δικτύων από τις νόθες καταστάσεις μπορεί να προκαλέσει θεραπεία νεύρωσης. Γίνονται ακόμα υποθέσεις που ισχυρίζονται ότι κάποιος βελτιώνει τη μνήμη του λησμονώντας τα νόθα πρότυπα σε ένα όνειρο.

### 2.2.3 Χωρητικότητα της Μνήμης

Αρχικά θα ασχοληθούμε με την ιδιότητα αναπαραγωγής (*reproduction property*) του δικτύου Hopfield. Η ιδιότητα αυτή εξαρτάται από το λόγο  $p/n$ , όπου  $p$  είναι ο αριθμός των προτύπων εκπαίδευσης και  $n$  ο αριθμός των νευρώνων. Το κλάσμα αυτό καθορίζει τη λεγόμενη χωρητικότητα (*capacity*) της αυτο - συσχετιζόμενης μνήμης, δεδομένου ότι όλα τα πρότυπα αποθηκεύονται χωρίς ανεπιθύμητα λάθη. Υποθέτουμε πως οι καταστάσεις των νευρώνων επιλέγονται ανεξάρτητα, με ίση πιθανότητα για τις διπολικές τιμές, τότε για αρκετά μεγάλα  $n$  και  $p$ , η πιθανότητα  $P$  μια κατάστασης να είναι σταθερή είναι [27]:

$$P = \frac{1}{2} - \frac{1}{\sqrt{\pi}} \int_0^{\sqrt{n/2p}} e^{-x^2} dx \quad (2.32)$$

Για παράδειγμα, περιμένουμε πως για  $p = 0.185n$  ο αριθμός των ασταθών καταστάσεων δεν υπερβαίνει το 1%. Ομως το αποτέλεσμα αυτό δεν εγγυάται ότι το δίκτυο θα φτάσει σε σταθερή κατάσταση. Μια παρόμοια ανάλυση [23, 27] (που ασχολείται με την αναπαραγωγή των περισσότερων ή όλων των πλήρων προτύπων), δείχνει ότι ο αριθμός των προτύπων

που μπορεί να αποθηκευτεί σε μια Hopfield μνήμη με  $n$  νευρώνες, είναι ασυμπτωτικά ανάλογη με  $n/\log n$ .

Μια πιο λεπτομερής ανάλυση [3, 4, 10, 24] δείχνει για  $p \leq 0.138n$  (δηλαδή για χωρητικότητα της μνήμης το πολύ 0.138), πως αυτά τα πρότυπα αντιστοιχούν στα τοπικά ελάχιστα της συνάρτησης ενέργειας στο δίκτυο Hopfield που έχει προσαρμοστεί με τον κανόνα του Hebb και άρα αυτό το δίκτυο μπορεί να χρησιμοποιηθεί ως αυτο - συσχετιζόμενη μνήμη. Αντίθετα, για  $p > 0.138n$  το τοπικό ελάχιστο εξαφανίζεται. Ακόμα, για  $p < 0.05n$  τα πρότυπα αντιστοιχούν στα ολικά ελάχιστα της συνάρτησης ενέργειας που είναι βαθύτερα από τα τοπικά των νόθων καταστάσεων. Αυτό σημαίνει ότι σε μια καλή μνήμη, χρειάζονται 200 νευρώνες για να αποθηκεύσουμε 10 πρότυπα, κάτι που περιλαμβάνει 40.000 συνδέσεις με (ακέραια) βάρη σε μια πλήρη τοπολογία κυκλικού δικτύου. Άρα το βασικό μοντέλο της μνήμης Hopfield, έχει προς το παρόν μόνο θεωρητική σημασία, εξαιτίας της χαμηλής του χωρητικότητας. Υπάρχουν αρκετές τροποποιήσεις που προσπαθούν να ξεπεράσουν αυτό το μειονέκτημα.

#### 2.2.4 Παράδειγμα Εφαρμογής ενός Hopfield

Εφαρμόζουμε τις παραπάνω γενικές αρχές σε ένα παράδειγμα για ελάττωση του θορύβου στην αναγνώριση χαρακτήρων. Το κάθε ψηφίο μπορεί να αναπαρασταθεί από έναν πίνακα διαστάσεων  $12 \times 10$  με άσπρα και μαύρα pixels, που αντιστοιχούν σε νευρώνες του δικτύου, έτσι ώστε οι καταστάσεις 1 και -1 να αναπαριστούν το μαύρο και άσπρο, αντίστοιχα.

Δείγματα από οκτώ ψηφία δημιουργήθηκαν και χρησιμοποιήθηκαν σε ένα δίκτυο Hopfield που χρησιμοποιούσε τον κανόνα Hebb. Παρουσιάστηκε στο δίκτυο μια εικόνα του "3" εισάγοντας 25 (σχήμα 2.4) να απομακρύνει το θόρυβο, λειτούργησε δηλαδή ως συσχετιζόμενη μνήμη αφού ανακατασκεύασε την αρχική εικόνα του "3" (που αντιστοιχεί στην τελική κατάσταση του δικτύου).

### 2.3 Συνεχές Δίκτυο Hopfield

Σε αυτήν την παράγραφο θα περιγράψουμε μια συνεχή παραλλαγή ενός αναλογικού δικτύου Hopfield [16] και την εφαρμογή του στην ευριστική

#### Σχήμα 2.4: Παράδειγμα εφαρμογής ενός δικτύου Hopfield

επίλυση του προβλήματος του περιπλανόμενου πωλητή (*Traveling Salesman Problem - TSP*). Σε αντίθεση με τα αναλογικά μοντέλα των νευρωνικών δικτύων που έχουμε δει ως τώρα (π.χ. δίκτυα πολλών επιπέδων με εκπαίδευση *backpropagation*), όπου η πραγματική κατάσταση του νευρώνα είναι μια συνεχής συνάρτηση του επιπέδου διέγερσης (*excitation level*), αυτό το δίκτυο είναι ένα παράδειγμα ενός μοντέλου το οποίο είναι πραγματική συνάρτηση του χρόνου. Σε τέτοιες περιπτώσεις οι υπολογιστικές δυναμικές συνήθως δίνονται από διαφορικές εξισώσεις, οι λύσεις των οποίων δεν μπορούν να δοθούν σε σαφή μορφή. Επομένως, αυτά τα μοντέλα δεν είναι κατάλληλα για προσομοιώσεις σε παραδοσιακούς υπολογιστές. Έχουν όμως πλεονεκτήματα σε αναλογικές εφαρμογές από ηλεκτρικά κυκλώματα.

##### 2.3.1 Συνεχής Υπολογιστική Δυναμική

Ας υποθέσουμε πως ένα δίκτυο Hopfield έχει την ίδια αρχιτεκτονική και πιθανόν την ίδια δυναμική με αυτήν που περιγράφεται στην παράγραφο 2.2.1. Ακόμα, έστω ότι τα συμμετρικά βάρη του δικτύου μπορούν να

είναι πραγματικά και  $w_{j0}$  ( $1 \leq j \leq n$ ) μια, γενικά, μη - μηδενική πραγματική πόλωση (ένα κατώφλι με αντίθετο πρόσημο) του  $j$  νευρώνα, που αντιστοιχεί σε ένα input  $y_0 = 1$ . Υπενθυμίζουμε ότι  $w_{jj} = 0$  ( $j = 1, \dots, n$ ). Τότε θεωρούμε την ακόλουθη υπολογιστική δυναμική για το συνεχές Hopfield. Σε χρόνο  $\theta$ , οι καταστάσεις των νευρώνων έχουν πάρει τις τιμές του input του δικτύου  $x = (x_1, \dots, x_n)$ , δηλαδή  $y_i^{(0)} = x_i$  ( $i = 1, \dots, n$ ). Κατά τη διάρκεια του υπολογισμού, η εξέλιξη της κατάστασης του δικτύου  $y(t)$  είναι μια συνεχής συνάρτηση του χρόνου  $t > \theta$ , που δίνεται από το ακόλουθο σύστημα διαφορικών εξισώσεων (για λόγους απλότητας η παράμετρος του χρόνου  $t$  παραλείπεται):

$$\tau_j \frac{dy_j}{dt} = -y_j + \sigma(\xi_j) = -y_j + \sigma\left(\sum_{i=0}^n w_{ji}y_i\right) \quad (2.33)$$

όπου  $\tau_j > 0$  ( $j = 1, \dots, n$ ) κατάλληλες σταθερές του χρόνου,  $\xi_j(t)$  το πραγματικό επίπεδο διέγερσης του νευρώνα  $j$ , που επίσης εξαρτάται από το χρόνο, και  $\sigma$  η συνεχής συνάρτηση ενεργοποίησης, σιγμοειδής στο  $(0, 1)$ :

$$\sigma(\xi) = \frac{1}{1 + e^{-\lambda\xi}} \quad (2.34)$$

ή τη διπολική μορφή του στο  $(-1, 1)$ :

$$\sigma(\xi) = \text{tgh}\left(\frac{1}{2}\lambda\xi\right) = \frac{1 - e^{-\lambda\xi}}{1 + e^{-\lambda\xi}} \quad (2.35)$$

όπου  $\lambda > 0$  είναι η παράμετρος της απολαβής (gain parameter). Για  $\lambda \rightarrow \infty$ , τα διακριτά outputs προκύπτουν όπως ακριβώς και στο διακριτό Hopfield (2.28).

Ο υπολογισμός του συνεχούς Hopfield τερματίζεται σε  $t^*$  όταν το δίκτυο βρίσκεται σε σταθερή κατάσταση  $y(t^*)$  της οποίας η μεταβολή στο χρόνο είναι 0, δηλαδή  $\frac{dy_j}{dt}(t^*) = 0$ ,  $j = 1, \dots, n$ . Αφού αντικαταστήσουμε την παραπάνω εξίσωση στην (2.33) παίρνουμε:

$$-y_j = \sigma(\xi_j) = \sigma\left(\sum_{i=0}^n w_{ji}y_i\right) \quad (2.36)$$

Ακόμα, η υπολογιστική δυναμική (2.33) μπορεί απίσης να περιγραφεί από το ακόλουθο σύστημα:

$$\tau_j \frac{d\xi_j}{dt} = -\xi_j + \sum_{i=0}^n w_{ji} y_i = -\xi_j + \sum_{i=0}^n w_{ji} \sigma(\xi_i) \quad (2.37)$$

Κάτω από ορισμένες προϋποθέσεις, το σύστημα (2.37) έχει τις ίδιες εξισώσεις ισορροπίας (2.36) με το (2.33).

Ανάλογα με τη διακριτή περίπτωση, η ενέργεια του συνεχούς Hopfield καθορίζεται:

$$E(t) = -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ji} y_j y_i - \sum_{j=1}^n w_{j0} y_j + \sum_{j=1}^n \int_0^{y_j} \sigma^{-1}(y) dy \quad (2.38)$$

Για τη συνάρτηση ενεργοποίησης (2.34) (όμοια και για την (2.35)), τα ολοκληρώματα της σχέσης (2.38) μπορούν αναλυτικά να γραφτούν:

$$\int_0^{y_j} \sigma^{-1}(y) dy = \frac{1}{\lambda} \ln y_j^{y_j} (1-y)^{1-y_j} \quad j = 1, \dots, n. \quad (2.39)$$

Θα δείξουμε πως η συνάρτηση ενέργειας  $E(t)$  είναι μη - αύξουσα στο υπολογιστικό σχήμα (2.33):

$$\frac{dE}{dt} = -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ji} \frac{dy_j}{dt} y_i - \frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ji} y_j \frac{dy_i}{dt} - \sum_{j=1}^n w_{j0} \frac{dy_j}{dt} + \sum_{j=1}^n \sigma^{-1}(y_j) \frac{dy_j}{dt}. \quad (2.40)$$

Από το γεγονός ότι τα βάρη είναι συμμετρικά και από τις (2.36), (2.37), παίρνουμε:

$$\frac{dE}{dt} = -\sum_{j=1}^n \frac{dy_j}{dt} \left( \sum_{i=1}^n w_{ji} y_i - \xi_j \right) = -\sum_{j=1}^n \tau_j \frac{dy_j}{dt} \frac{d\xi_j}{dt} \quad (2.41)$$

Η παράγωγος  $\frac{dE}{dt}$  μπορεί να εκφραστεί από την (2.36) με τέτοιο τρόπο ώστε να δούμε ότι είναι μη - θετική:

$$\frac{dE}{dt} = -\sum_{j=1}^n \tau_j \sigma'(\xi_j) \left( \frac{d\xi_j}{dt} \right)^2 \leq 0 \quad (2.42)$$

αφού  $\tau_j > 0$  ( $j = 1, \dots, n$ ) και η συνάρτηση ενεργοποίησης  $\sigma(\xi)$  είναι αύξουσα για  $\lambda > 0$ , δηλαδή  $\sigma'(\xi_j) > 0$ . Άρα η ενέργεια  $E(t)$  μειώνεται

κατά τη φάση του υπολογισμού (2.33), δηλαδή  $\frac{dE}{dt} < 0$  έως ότου το δίκτυο να φτάσει σε τοπικό ελάχιστο, όπου  $\frac{dE}{dt} = 0$ , αφού η  $E(t)$  είναι φραγμένη (παίρνει τιμές στο  $(0, 1)$  ή στο  $(-1, 1)$  και τα αντίστοιχα ολοκληρώματα (2.39) είναι φραγμένα). Αρα το συνεχές Hopfield αποδεικνύεται ότι τερματίζει πάντα τον υπολογισμό του σε μια σταθερή κατάσταση.

Σε αναλογία με τη διακριτή περίπτωση, το συνεχές Hopfield που έχει προσαρμοστεί με τον κανόνα του Hebb, μπορεί να χρησιμοποιηθεί ως αυτο - συσχετιζόμενη μνήμη.

### 2.3.2 Το Πρόβλημα του Περιπλανόμενου Πωλητή (TSP)

Αποδείξαμε ότι το συνεχές Hopfield ελαχιστοποιεί τη συνάρτηση ενέργειας  $E$  κατά τη φάση του υπολογισμού. Αυτό μπορεί να χρησιμοποιηθεί για την ευριστική επίλυση ενός προβλήματος βελτιστοποίησης, που μπορεί η συνάρτησή του μαζί με τους περιορισμούς της να εκφραστεί ως ένα "δευτεροβάθμιο πολυώνυμο" (2.38). Συγκρίνοντας την αντικειμενική συνάρτηση με τη συνάρτηση ενέργειας, τα βάρη του συνεχούς Hopfield αναπτύσσονται (φάση προσαρμογής). Τότε η βέλτιστη και εφικτή λύση ενός δοθέντος προβλήματος αναζητείται στη φάση του υπολογισμού. Αυτή η προσέγγιση θα εξεταστεί για το πρόβλημα του περιπλανόμενου πωλητή (TSP) [17].

Εστω  $N > 2$  ο αριθμός των πόλεων και για κάθε ζευγάρι  $1 \leq r, s \leq N$  ορίζουμε τη συνάρτηση  $d : \{1, \dots, N\}^2 \rightarrow R$  που ορίζει την απόσταση  $d(r, s)$  από την πόλη  $r$  στην  $s$ . Υποθέτουμε ακόμα πως  $d(r, r) = 0$  και  $d(r, s) = d(s, r)$  για κάθε  $1 \leq r, s \leq N$ . Τότε το πρόβλημα έγκειται στο να βρεθεί η σειρά των πόλεων που σχηματίζει το συντομότερο κυκλικό μονοπάτι (διαδρομή) έτσι ώστε ο πωλητής να έχει επισκεφτεί μόνο μια φορά την κάθε πόλη.

Το πρόβλημα μπορεί να αναπαρασταθεί από ένα συνεχές δίκτυο Hopfield με  $n = N \times N$  νευρώνες των οποίων οι καταστάσεις  $y_{ru}$  έχουν χωριστεί στα εξής ζεύγη: ο αριθμός της πόλης είναι  $r$  ( $1 \leq r \leq N$ ) και η θέση του στη διαδρομή είναι  $u$  ( $1 \leq r \leq N$ ). Στο σχήμα (2.5) βλέπουμε την τοπολογία του δικτύου, όπου οι σειρές αντιστοιχούν στις πόλεις και οι στήλες καθορίζουν τη σειρά που ο πωλητής σταματά σε αυτές.

Για παράδειγμα, ως συνάρτηση ενεργοποίησης παίρνουμε τη σιγμοειδή στο διάστημα  $(0, 1)$ :

Σχήμα 2.5: Η αναπαράσταση του δικτύου για το TSP

$$\sigma(\xi) = \frac{1}{1 + e^{-\lambda\xi}}, \quad (2.43)$$

ή τη διπολική της μορφή, στο διάστημα  $(-1,1)$ :

$$\sigma(\xi) = \frac{1 - e^{-\lambda\xi}}{1 + e^{-\lambda\xi}}, \quad (2.44)$$

όπου  $\lambda > 0$  είναι η παράμετρος του κέρδους (gain parameter). Για  $\lambda \rightarrow \infty$ , τα διακριτά αποτελέσματα  $-1$  ή  $1$  αποκτώνται όπως στο διακριτό δίκτυο Hopfield.

Δηλαδή οι καταστάσεις του νευρώνα  $y_{ru} \in (0,1)$  ( $1 \leq r, u \leq N$ ). Αν θεωρήσουμε ένα συνεχές δίκτυο Hopfield, σε χρόνο  $0$ , οι καταστάσεις των νευρώνων του προσδιορίζονται σύμφωνα με την είσοδο (input) του δικτύου  $x = (x_1, \dots, x_n)$ , δηλαδή  $y_i^{(0)} = x_i$  ( $i = 1, \dots, n$ ).

Όταν ελαχιστοποιείται η συνάρτηση ενέργειας (2.38), αναγκάζει τους νευρώνες να συγκεντρώνονται σε σταθερές καταστάσεις, δηλαδή  $y_{ru} \doteq 1$  ή  $y_{ru} \doteq 0$ . Η κατάσταση  $y_{ru} \doteq 1$  (ενεργός νευρώνας δηλαδή) σημαίνει ότι η  $r$  πόλη βρίσκεται στη  $u$  θέση της διαδρομής, αντίστοιχα για  $y_{ru} \doteq 0$  (παθητικός νευρώνας).

Προφανώς, σύμφωνα με αυτόν τον τρόπο αναπαράστασης του προβλήματος, δεν ανταποκρίνονται όλες οι καταστάσεις των νευρώνων σε πραγματικές διαδρομές. Πρέπει δηλαδή να λάβουμε υπόψη και το κατά πόσο η λύση είναι εφικτή. Αρχικά ο πωλητής πρέπει να πάει σε κάθε πόλη τουλάχιστον μια φορά, κάτι που σημαίνει ότι σε κάθε σειρά ένας το πολύ νευρώνας πρέπει να είναι ενεργός. Αυτό αντιστοιχεί στην ελαχιστοποίηση της ακόλουθης έκφρασης:

$$\begin{aligned} E_A &= \frac{A}{2} \sum_{r=1}^N \sum_{u=1}^N \sum_{\substack{v=1 \\ v \neq u}}^N y_{ru} y_{rv} \\ &= -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n -A\delta_{rs} (1 - \delta_{uv}) y_j y_i \quad (2.45) \\ &\quad j = (r, u) \quad i = (s, v) \end{aligned}$$



όπου η παράμετρος  $A > 0$  μετράει την επίδραση της  $E_A$  κατά τη διαδικασία της ελαχιστοποίησης, και το  $\delta_{rs}$  είναι το δέλτα του Kronecker, δηλαδή  $\delta_{rs} = 1$  αν  $r = s$  και  $\delta_{rs} = 0$  αν  $r \neq s$ .

Όμοια, ο πωλητής μπορεί να περάσει από μια το πολύ πόλη σε κάθε στάση της διαδρομής, δηλαδή σε κάθε στήλη το πολύ ένας νευρώνας πρέπει να είναι ενεργός (όπως φαίνεται και στο σχήμα (2.1)).

$$\begin{aligned}
 E_B &= \frac{B}{2} \sum_{u=1}^N \sum_{r=1}^N \sum_{\substack{s=1 \\ s \neq r}}^N y_{ru} y_{su} \\
 &= -\frac{1}{2} \sum_{\substack{j=1 \\ j=(r,u)}}^n \sum_{\substack{i=1 \\ i=(s,v)}}^n -B \delta_{uv} (1 - \delta_{rs}) y_j y_i \quad (2.46)
 \end{aligned}$$

όπου η παράμετρος  $B > 0$  μετράει την επίδραση της  $E_B$  κατά την ελαχιστοποίηση.

Τέλος, ο πωλητής πρέπει να περάσει από  $N$  πόλεις ακριβώς, δηλαδή  $N$  νευρώνες πρέπει να είναι ενεργοί στο δίκτυο (σχήμα (2.1)). Αυτό αντιστοιχεί στην ελαχιστοποίηση της έκφρασης:

$$\begin{aligned}
 E_C &= \frac{C}{2} \left( N - \sum_{r=1}^N \sum_{u=1}^N y_{ru} \right)^2 \quad (2.47) \\
 &= \frac{CN^2}{2} - \frac{1}{2} \sum_{\substack{j=1 \\ j=(r,u)}}^n \sum_{\substack{i=1 \\ i=(s,v)}}^n -C (1 - \delta_{ji}) y_j y_i - \sum_{j=1}^n \left( CN - \frac{C}{2} \right) y_j,
 \end{aligned}$$

όπου η παράμετρος  $C > 0$  μετράει την επίδραση της  $E_C$  κατά την ελαχιστοποίηση. Η ταυτόχρονη ελαχιστοποίηση των τριών τελευταίων σχέσεων στο χώρο καταστάσεων αναγκάζει τις καταστάσεις του δικτύου να παρουσιάσουν εφικτές λύσεις του προβλήματος.

Ακόμα, απαραίτητη προϋπόθεση είναι η διαδρομή να είναι όσο το δυνατό πιο σύντομη, άρα η λύση να είναι βέλτιστη. Αυτό ισοδυναμεί με την ελαχιστοποίηση της έκφρασης:

$$\begin{aligned}
E_D &= \frac{D}{2} \sum_{r=1}^N \sum_{s=1}^N \sum_{u=1}^N d(r,s) y_{ru} (y_{s,u-1} + y_{s,u+1}) \quad (2.48) \\
&= -\frac{1}{2} \sum_{\substack{j=1 \\ j=(r,u)}}^n \sum_{\substack{i=1 \\ i=(s,v)}}^n -Dd(r,s) (\delta_{u,v-1} - \delta_{u,v+1}) y_j y_i
\end{aligned}$$

όπου η παράμετρος  $D > 0$  μετράει την επίδραση της  $E_B$  κατά την ελαχιστοποίηση. Για λόγους απλότητας, οι δείκτες  $u - 1$  για  $u = 1$  και  $u + 1$  για  $u = N$  ερμηνεύονται ως  $u = N$  και  $u - 1$ , αντίστοιχα.

Αρα, η τελική συνάρτηση προς ελαχιστοποίηση  $E_{TS}$  είναι το άθροισμα των τεσσάρων τελευταίων εκφράσεων και το ελάχιστό της δίνει μια βέλτιστη εφικτή λύση για το πρόβλημα του περιπλανόμενου πωλητή. Επομένως η σειρά των πόλεων στη συντομότερη διαδρομή, δίνεται από την έκφραση:

$$\begin{aligned}
E_{TS} &= E_A + E_B + E_C + E_D = \\
&-\frac{1}{2} \sum_{\substack{j=1 \\ j=(r,u)}}^n \sum_{\substack{i=1 \\ i=(s,v)}}^n - \left( \begin{aligned} &-A\delta_{rs}(1 - \delta_{uv}) - B\delta_{uv}(1 - \delta_{rs}) - \\ &C(1 - \delta_{ji}) - Dd(r,s)(\delta_{u,v-1}\delta_{u,v+1}) \end{aligned} \right) y_j y_i \\
&\quad - \sum_{j=1}^n \left( CN - \frac{C}{2} \right) y_j. \quad (2.49)
\end{aligned}$$

Συγκρίνοντας την (2.49) με τη συνάρτηση ενέργειας, παίρνουμε τα βάρη ενός συνεχούς δικτύου Hopfield:

$$\begin{aligned}
w_{ji} &= \frac{-A\delta_{rs}(1 - \delta_{uv}) - B\delta_{uv}(1 - \delta_{rs}) -}{C(1 - \delta_{ji}) - Dd(r,s)(\delta_{u,v-1}\delta_{u,v+1})} \quad (2.50) \\
w_{j0} &= CN - \frac{C}{2} \quad j = (r,u), \quad i = (s,v), \quad 1 \leq r,s,u,v \leq N
\end{aligned}$$

Προκύπτει από την (2.50) ότι  $w_{ji} = w_{ij}$  και  $w_{jj} = 0 \quad 1 \leq i,j \leq n$ .

Για ευριστική επίλυση του προβλήματος, τα βάρη του συνεχούς δικτύου προσδιορίζονται σύμφωνα με τον τύπο (2.50). Εδώ, οι κατάλληλες πραγματικές παράμετροι  $A, B, C, D$  επιλέγονται ώστε να καθορίσουν το βαθμό της ελαχιστοποίησης με σεβασμό στους συγκεκριμένους

όρους της  $E_{TS} = E_A + E_B + E_C + E_D$  (πχ.  $A=B=D=500, C=200$ )<sup>2</sup>. Ο υπολογισμός αρχίζει με μια τυχαία αρχική κατάσταση, κοντά στο μηδέν (πχ.  $y_j^{(0)} \in (0, 0.001)$ ). Ο υπολογισμός συνεχίζεται μέχρι το δίκτυο να φτάσει σε μια σταθερή κατάσταση που αντιστοιχεί σε κάποιο ελάχιστο της συνάρτησης ενέργειας που συμπίπτει με την αντικειμενική συνάρτηση  $E_{TS}$  του προβλήματος αυτού. Τελικά το αποτέλεσμα που διαβάζεται από το δίκτυο, για παράδειγμα  $y_{ru} \doteq 1$  ερμηνεύεται ως εξής: ο πωλητής επισκέπτεται την  $r$  πόλη κατά την  $u$  στάση του στη διαδρομή.

Όμως δεδομένου ότι το ελάχιστο της συνάρτησης μπορεί να μην είναι ολικό, το αποτέλεσμα που θα πάρουμε μπορεί να μην είναι βέλτιστο και μπορεί να μην είναι καν εφικτό. Με κατάλληλη επιλογή των παραμέτρων  $A, B, C, D$  και με διαφορετικές αρχικές συνθήκες στο δίκτυο, ίσως έχουμε καλύτερη προσέγγιση. Επίσης μπορεί να βοηθήσει μια καλή στρατηγική επιλογής της παραμέτρου  $\lambda$ . Αρχικά επιλέγεται πολύ μικρό, προκειμένου να δώσει στις καταστάσεις των νευρώνων μεγαλύτερο βαθμό ελευθερίας. Όταν, κατά τη φάση του υπολογισμού, το δίκτυο πλησιάζει στο ελάχιστο της συνάρτησης, η παράμετρος μπορεί να αυξηθεί ώστε να επιταχύνει τη σύγκλιση. Παρ' όλα αυτά, δεν μπορούν να δοθούν σαφείς οδηγίες για την επιλογή αυτών των παραμέτρων, ώστε να βεβαιωθούμε για σύγκλιση σε ολικό ελάχιστο της συνάρτησης ενέργειας, προκειμένου να πετύχουμε την πραγματικά βέλτιστη λύση στο συγκεκριμένο πρόβλημα, αφού πρόκειται για ένα πρόβλημα *NP-complete*. Η διαδικασία που περιγράφηκε, είναι απλά μια ευριστική προσέγγιση που δεν ανταγωνίζεται τους γνωστούς, κλασικούς αλγόριθμους για το πρόβλημα. Μας δείχνει όμως μια πιθανή, μη συνηθισμένη εφαρμογή των νευρωνικών δικτύων.

## 2.4 Μηχανή Boltzmann

Μια μηχανή Boltzmann είναι ένα μοντέλο νευρωνικού δικτύου, που εισήγαγαν οι Hinton και Sejnowski [1, 29, 30]. Πρόκειται για μια στοχαστική παραλλαγή του δικτύου Hopfield με κρυμμένους νευρώνες. Το όνομα του μοντέλου έχει τις ρίζες του στη στατιστική μηχανική, επειδή

<sup>2</sup>J.J.Hopfield and D.W.Tank, "Neural" computation of decisions in optimization problems, Biological Cybernetics, 52:141-152,1985

Σχήμα 2.6: Παράδειγμα της τοπολογίας της μηχανής Boltzmann

η πιθανότητα των καταστάσεών του στη λεγόμενη θερμική ισορροπία ελέγχεται από την κατανομή Boltzmann. Για παράδειγμα, η μηχανή Boltzmann μπορεί να χρησιμοποιηθεί ως ετερο - συσχετιζόμενη μνήμη.

#### 2.4.1 Στοχαστική Υπολογιστική Δυναμική

Η αρχιτεκτονική δυναμική της μηχανής Boltzmann καθορίζει μια τοπολογία κυκλικού νευρωνικού δικτύου με συμμετρικές συνδέσεις, δηλαδή μπορεί να περιγραφεί από ένα μη - κατευθυνόμενο γράφο. Το σύνολο  $V = A \cup B$  των  $s$  νευρώνων χωρίζεται σε ένα σύνολο  $B$  από  $b$  κρυμμένους νευρώνες και  $A = X \cup Y$  από  $a = n + m$  που καλούνται ορατοί νευρώνες και κατηγοριοποιούνται σε ένα σύνολο  $X$  από  $n$  inputs νευρώνες και σε ένα σύνολο  $Y$  από  $m$  output νευρώνες. Ένα παράδειγμα μιας αρχιτεκτονικής της μηχανής Boltzmann φαίνεται στο σχήμα (2.6).

Οι νευρώνες σημειώνονται με δείκτες  $i, j$  κλπ, με  $\xi_j$  συμβολίζει το πραγματικό excitation level και το  $y_j \in \{-1, 1\}$  μια διπολική κατάσταση

του νευρώνα  $j$ . Οι καταστάσεις των ορατών νευρώνων δηλώνονται με ένα διάνυσμα  $\alpha \in \{-1, 1\}^a$  και αντίστοιχα  $\beta \in \{-1, 1\}^b$  των κρυμμένων, αντίστοιχα. Τότε με  $y = (\alpha, \beta) \in \{-1, 1\}^s$  δηλώνουμε την κατάσταση της μηχανής Boltzmann. Η σύνδεση μεταξύ  $i$  και  $j$  παίρνει ένα πραγματικό συμμετρικό βάρος  $w_{ji} = w_{ij}$ . Για λόγους απλότητας θεωρούμε τα κατώφλια όλων των νευρώνων ίσα με μηδέν, και  $w_{jj} = 0 \forall j \in V$ . Επίσης χρησιμοποιούμε το συμβολισμό  $\bar{j}$  για τους νευρώνες που συνδέονται με το νευρώνα  $j$ .

Η υπολογιστική δυναμική της μηχανής Boltzmann θα περιγραφεί για την περίπτωση σειριακών υπολογισμών. Αρχικά, σε χρόνο  $0$ , στις καταστάσεις των input νευρώνων  $i \in X$  δίνονται τιμές από ένα διπολικό input του δικτύου  $x = (x_1, \dots, x_n) \in \{-1, 1\}^n$  και αυτά κατόπιν, καθορίζονται κατά τον υπολογισμό, δηλαδή  $y_i^t = x_i$  για  $i \in X$ ,  $t \geq 0$ . Οι καταστάσεις  $y_j^0 \in \{-1, 1\}$  των νευρώνων  $j \in V \setminus X$  (non - input νευρώνες) αρχικοποιούνται τυχαία. Στη συνέχεια, επιλέγονται τυχαία ένας από αυτούς οι νευρώνες και οι καταστάσεις του αναβαθμίζονται, ενώ οι υπόλοιποι νευρώνες μένουν αμετάλλαχτοι. Όπως και στο δίκτυο Hopfield, θεωρούμε ξανά το μακροσκοπικό χρόνο  $\tau$ , δηλαδή  $\tau s' < (\tau + 1) s'$  όπου  $s' = s - n$ , ο αριθμός των αναβαθμισμένων (non - input) νευρώνων κατά τη διάρκεια μιας περιόδου. Αρχικά, υπολογίζεται ένα πραγματικό επίπεδο διέγερσης του νευρώνα  $j$ :

$$\xi_j^{(t-1)} = \sum_{i \in \bar{j}} w_{ji} y_i^{(t-1)} \quad (2.51)$$

Τότε το output του νευρώνα  $j$  σε χρόνο  $t$  ορίζεται στοχαστικά, έτσι ώστε ο νευρώνας  $j$  να είναι ενεργός με πιθανότητα:

$$P\{y_j^{(t)} = 1\} = \sigma(\xi_j^{(t-1)}) \quad (2.52)$$

και, επομένως είναι παθητικός με πιθανότητα:

$$P\{y_j^{(t)} = -1\} = 1 - P\{y_j^{(t)} = 1\} = \sigma(-\xi_j^{(t-1)}) \quad (2.53)$$

όπου  $\sigma$  είναι μια στοχαστική συνάρτηση ενεργοποίησης:

$$\sigma(\xi) = \frac{1}{1 + e^{-2\xi/T\tau}} \quad (2.54)$$

και το σχήμα της αντιστοιχεί στη σιγμοειδή (1.9).

## 2.4.2 Simulated Annealing

Η παράμετρος  $T^\tau$  στον τύπο (2.54) καλείται θερμοκρασία χάρη στην υλική αναλογία. Είναι αντιστρόφως ανάλογη με την παράμετρο της στοχαστικής συνάρτησης ενεργοποίησης. Για θερμοκρασία  $T \rightarrow \infty$ , οι πιθανότητες (2.52), (2.53) και των δύο διπολικών καταστάσεων είναι  $1/2$  και η μηχανή Boltzmann συμπεριφέρεται τελείως τυχαία κατά τον υπολογισμό. Αντιθέτως, για  $T \rightarrow 0$  ο υπολογισμός της μηχανής Boltzmann με τις (2.51), (2.52) είναι ντετερμινιστικός και συμπίπτει με την υπολογιστική φάση του δικτύου Hopfield που περιγράφεται από τις εξισώσεις (2.27), (2.28). Συνήθως επιλέγεται αρκετά υψηλή θερμοκρασία  $T^{(0)}$ , έτσι ώστε η πιθανότητα (2.52) να είναι λίγο μεγαλύτερη από  $1/2$  και ο υπολογισμός έχει μεγάλο βαθμό ελευθερίας. Μετά, λόγω της σύγκλισης, όλο το σύστημα τελικά "παγώνει", δηλαδή η θερμοκρασία σιγά-σιγά μειώνεται. Αυτή η διαδικασία καλείται *simulated annealing* [63].

Για παράδειγμα η θερμοκρασία  $T^\tau = n^c T^0$  για  $cq \leq \tau < (c+1)q$  ( $c = 0, 1, \dots$ ) μειώνεται μετά από κάθε  $q > 0$  μακροσκοπικά βήματα, όπου  $0 < n < 1$  επιλέγεται κοντά στη μονάδα και είναι αντιστρόφως ανάλογο με την επιλογή του  $q$  (πχ.  $n = 0.98$  για μικρό  $q$  και  $n = 0.9$  για μεγαλύτερο  $q$ ). Η εξέλιξη της θερμοκρασίας ελέγχεται από το ακόλουθο ευριστικό :

$$T^\tau = \frac{T^0}{\log(1 + \tau)} \quad \tau > 0 \quad (2.55)$$

Το γράφημα της στοχαστικής συνάρτησης ενεργοποίησης φαίνεται στο σχήμα 2.7.

Δηλαδή οι περιστροφές (*spins*) στα μαγνητικά υλικά επηρεάζονται από τυχαίες θερμικές, όπως λέγονται, ταλαντώσεις. Αυτές περιορίζουν την επιρροή του περιβάλλοντος μαγνητικού πεδίου, ενώ ο λόγος των επιδράσεων εξαρτάται από τη θερμοκρασία. Οι θερμικές ταλαντώσεις κυριαρχούν έως το απόλυτο μηδέν και τότε εξαφανίζονται. Αντίθετα, υπερέχουν από υψηλές θερμοκρασίες και ο μαγνητικός προσανατολισμός μιας περιστροφής είναι ανεξάρτητος του μαγνητικού πεδίου. Αυτό το φαινόμενο περιγράφεται από την Glauber δυναμική [11] που αντιστοιχεί στην υπολογιστική δυναμική της μηχανής Boltzmann. Η ελάττωση της θερμοκρασίας προκαλεί την πραγματική *material annealing*.

Η υπολογιστική δυναμική της μηχανής Boltzmann μπορεί να εξηγη-

Σχήμα 2.7: Το γράφημα της στοχαστικής συνάρτησης ενεργοποίησης για διαφορετικές θερμοκρασίες  $T$

θεί από τη νευροφυσιολογία. Εκτός από τα κανονικά ηλεκτρικά σήματα από τους δενδρίτες ο βιολογικός νευρώνας επηρεάζεται από άλλες συνθήκες. Για παράδειγμα, η ώθηση που δημιουργείται σε ένα νευράξονα (αχον) έχει διαφορετική ένταση κάθε φορά, υπάρχουν καθυστερήσεις στις συνάψεις, τυχαίες ταλαντώσεις κλπ. Όλα αυτά μπορούν να θεωρηθούν ως θόρυβος που παρουσιάζεται από τις θερμικές ταλαντώσεις. Σε αυτήν την περίπτωση, η παράμετρος  $T$  δεν αντιστοιχεί σε πραγματική θερμοκρασία αλλά καθορίζει το επίπεδο του θορύβου.

### 2.4.3 Κατάσταση Ισορροπίας

Όπως και στο Hopfield η συνάρτηση ενέργειας της μηχανής Boltzmann καθορίζεται από τον τύπο:

$$E(y) = -\frac{1}{2} \sum_{j \in V} \sum_{i \in j} w_{ji} y_j y_i \quad (2.56)$$

Μπορεί να αποδειχθεί ότι η ενέργεια (2.56) ελαττώνεται ανάλογα με την παράμετρο θερμοκρασίας  $T$  (simulated annealing) κατά τη φάση του υπολογισμού, έως ότου σε χρόνο  $t^*$ , η μηχανή Boltzmann να φτάσει τη λεγόμενη θερμική ισορροπία (thermal equilibrium) με θερμοκρασία  $T^*$ . Τότε η κατάσταση της μηχανής Boltzmann δεν είναι σταθερή λόγω

του στοχαστικού υπολογισμού, αλλά ταλαντώνεται τοπικά γύρω από τη μέση "σταθερή" κατάσταση  $\bar{y}^* \in [-1, 1]^s$  που αντιστοιχεί στο τοπικό ελάχιστο της  $E$ . Ένα θεμελιώδες αποτέλεσμα της μηχανικής λέει πως σε θερμική ισορροπία, με θερμοκρασία  $T^*$ , κάθε μια από τις πιθανές καταστάσεις  $y^* \in [-1, 1]^s$  της μηχανής Boltzmann συμβαίνει με πιθανότητα (κατανομή Boltzmann):

$$p_B(y)^* = \frac{e^{-E(y^*)/T^*}}{\sum_{y \in \{-1, 1\}^s} e^{-E(y^*)/T^*}} \quad (2.57)$$

Με τη σχέση (2.57) η σταθερή κατάσταση ισορροπίας  $\bar{y}^* \in [-1, 1]^s$  που είναι ένας μέσος των καταστάσεων στη θερμική ισορροπία, υπολογίζεται ως εξής:

$$\bar{y}^* = \sum_{y \in \{-1, 1\}^s} p_B(y) y_j \quad j \in V \quad (2.58)$$

Τότε ο μέσος των καταστάσεων των νευρώνων  $\bar{y}^* \in [-1, 1]^s$  μπορεί να καθορίσει το μέσο output για δοσμένο input. Εκτός από ετεροσυσχετιζόμενη μνήμη, η μηχανή Boltzmann μπορεί να χρησιμοποιηθεί για ευριστική επίλυση εργασιών βελτιστοποίησης, δεδομένου ότι ένα πρόβλημα βελτιστοποίησης με την αντίστοιχη αντικειμενική συνάρτηση και τους περιορισμούς μπορεί να μετατραπεί σε ελαχιστοποίηση μιας τετραγωνικής μορφής. Σε αυτήν την περίπτωση τα βάρη της μηχανής Boltzmann τα παίρνουμε συγκρίνοντας την αντίστοιχη τετραγωνική μορφή με τη συνάρτηση ενέργειας  $E$ . Μια εφικτή και βέλτιστη λύση του δοθέντος προβλήματος αναζητείται κατά τη φάση του υπολογισμού ελαχιστοποιώντας τη συνάρτηση ενέργειας χρησιμοποιώντας μια "στοχαστική μέθοδο κλίσης". Το κυριότερο πρόβλημα στη μη - γραμμική βελτιστοποίηση είναι να αποφύγουμε τα τοπικά ελάχιστα προκειμένου να φτάσουμε σε ένα ολικό ελάχιστο. Το πλεονέκτημα της μηχανής Boltzmann έναντι του Hopfield έγκειται στην πιθανότητα, κάτω από υψηλή θερμοκρασία, να μετακινήθουμε από ένα τοπικό ελάχιστο της  $E$  σε μια κατάσταση με υψηλότερη ενέργεια (χάρη στο στοχαστικό υπολογισμό) από την οποία ένα μονοπάτι στο χώρο καταστάσεων οδηγεί σε ένα ελάχιστο. Χρησιμοποιώντας κατάλληλη στρατηγική στο simulated annealing, μπορεί να βρεθεί ένα καλύτερο βέλτιστο από το ντετερμινιστικό Hopfield. Επίσης, στην περίπτωση της αυτο-συσχετιζόμενης



μνήμης όταν τα ολικά ελάχιστα της  $E$  παρουσιάζουν τα πρότυπα εκπαίδευσης, ενώ τα τοπικά αντιστοιχούν στα νόθα (spurious) πρότυπα. Αυτές οι καταστάσεις μπορούν να αποφευχθούν με το στοχαστικό τρόπο υπολογισμού.

## 2.4.4 Μάθηση Boltzmann

Σε αυτήν την παράγραφο θα περιγράψουμε την προσαρμοστική δυναμική της μηχανής Boltzmann για την περίπτωση της ετερο - συσχετιζόμενης μνήμης. Η επιθυμητή συνάρτηση του δικτύου είναι προκαθορισμένη από μια διακριτή πιθανότητα κατανομής καταστάσεων ορατών νευρώνων έτσι ώστε μια επιθυμητή πιθανότητα  $p_d(x, d) > 0$  να προσδιοριστεί σε κάθε πιθανή κατάσταση  $\alpha = (x, d) \in \{-1, 1\}^{n+m}$  των ορατών νευρώνων. Στην πράξη όμως μια τέτοια κατανομή είναι συνήθως άγνωστη και επίσης η περιγραφή της θα απαιτούσε εκθετικά πολλές ( $2^\alpha$ ,  $\alpha = m + n$ ) μη-μηδενικές τιμές. Αρα αντί για αυτό θεωρείται ένα *training set*:

$$T = \left\{ (x_k, d_k) \mid x_k = (x_{k1}, \dots, x_{kn}) \in \mathfrak{R} \right. \\ \left. d_k = (d_{k1}, \dots, d_{km}) \in \{0, 1\}^m \quad k = 1, \dots, p \right\} \quad (2.59)$$

που περιέχει μόνο τα σχετικά inputs  $x_k$  ( $k = 1, \dots, p$ ) με τα επιθυμητά outputs  $d_k$ , έτσι ώστε ο αριθμός  $0 < p \ll 2^\alpha$  των προτύπων να είναι πολύ μικρότερος από τον αριθμό όλων των δυνατών καταστάσεων των ορατών νευρώνων. Ακόμα, μια ομοιόμορφη κατανομή από πρότυπα εκπαίδευσης έχει υποτεθεί, δηλαδή  $p_d(x_k, d_k) = \frac{1}{p} > 0$   $k = 1, \dots, p$ . Ακόμα, ένας τυχαίος θόρυβος με μικρή πιθανότητα μεταφέρεται στο σύνολο εκπαίδευσης  $T$ , που διαβεβαιώνει ότι οι υπόλοιπες καταστάσεις των ορατών νευρώνων που δεν περιλαμβάνονται σε αυτό, συμβαίνουν με μη-μηδενικές πιθανότητες.

Ο σκοπός της προσαρμογής της μηχανής Boltzmann είναι να βρούμε μια διαμόρφωση του δικτύου  $\mathbf{w}$  ώστε η κατανομή πιθανότητας  $p_A(\alpha)$  των καταστάσεων  $\alpha \in \{-1, 1\}^\alpha$  των ορατών νευρώνων από το  $A = X \cup Y$  στη θερμική ισορροπία συμπίπτει με την επιθυμητή κατανομή  $p_d(\alpha)$  των καταστάσεων  $\alpha = (x, d)$  στα πρότυπα (2.59). Η πιθανότητα  $p_A(\alpha)$  μιας κατάστασης  $\alpha \in \{-1, 1\}^\alpha$  ορατού νευρώνα μπορεί να εκφραστεί (μέσω

της (2.57) ανεξάρτητα από τις καταστάσεις στους κρυμμένους νευρώνες  $\beta \in \{-1, 1\}^b$  ως εξής:

$$p_A(\alpha) = \sum_{\beta \in \{-1, 1\}^b} p_B(\alpha, \beta) \quad (2.60)$$

Μια σχετική εντροπία που παίρνει βάρη από τις πιθανότητες των προτύπων εκπαίδευσης μετράει επαρκώς τη διαφορά μεταξύ των κατανομών πιθανοτήτων  $p(A)$  και  $p(d)$ , δηλαδή αναπαριστά ένα σφάλμα  $\epsilon$ :

$$\epsilon(w) = \sum_{\alpha \in \{-1, 1\}^a} p_d(\alpha) \log \frac{p_d(\alpha)}{p_A(\alpha)} \quad (2.61)$$

Μπορεί ναδειχτεί ότι το  $\epsilon(w)$  ισούται με μηδέν τότε και μόνο όταν  $p_A = p_d$ . Ακόμα, το  $\epsilon(w)$  είναι μια συνάρτηση της διαμόρφωσης  $w$  της μηχανής Boltzmann, αφού  $p_A$  εξαρτάται από τα συναπτικά βάρη, σύμφωνα με τις (2.60), (2.56) και (2.57). Αρα η συνάρτηση σφάλματος  $\epsilon(w)$  μπορεί να ελαχιστοποιηθεί στο χώρο των βαρών χρησιμοποιώντας μια μέθοδο κλίσης.

Στην αρχή της φάσης της προσαρμογής της μηχανής Boltzmann, σε χρόνο  $0$ , τα βάρη στη διαμόρφωση  $w^0$  είναι ομοιόμορφα κατανεμημένα γύρω από το  $0$ , δηλαδή  $w_{ji}^{(0)} = w_{ij}^{(0)} \in \langle -1, 1 \rangle$  ( $j \in V, i \in \bar{j}$ ). Η προσαρμογή προχωράει με διακριτά χρονικά βήματα που αντιστοιχούν στα training epochs στα οποία κάθε πρότυπο από το  $T$  παρουσιάζεται στο δίκτυο περισσότερες φορές. Η νέα διαμόρφωση υπολογίζεται για  $t > 0$ :

$$w_{ji}^{(t)} = w_{ji}^{(t-1)} + \Delta w_{ji}^{(t)} \quad j \in V, i \in \bar{j} \quad (2.62)$$

όπου  $\Delta w_{ji}^{(t)}$  ( $t > 0$ ) είναι ανάλογο με το αντίθετο της κλίσης της συνάρτησης σφάλματος στο σημείο  $w^{(t-1)}$ :

$$\Delta w_{ji}^{(t)} = -\epsilon \frac{\partial \epsilon}{\partial w_{ji}} (w^{(t-1)}) \quad j \in V, i \in \bar{j} \quad (2.63)$$

όπου  $0 < \epsilon < 1$  είναι ο ρυθμός μάθησης.

Για να εφαρμόσουμε την (2.62) πρέπει να υπολογίσουμε την κλίση της συνάρτησης σφάλματος στην (2.63). Διαφορίζοντας τη συνάρτηση

σφάλματος (2.61) ενώ εφαρμόζουμε τις (2.60), (2.57), και (2.56) και ερμηνεύοντας τις αντίστοιχες πιθανότητες, παίρνουμε την ακόλουθη έκφραση:

$$\frac{\partial \epsilon}{\partial w_{ji}} = -\frac{1}{T^*} \left( \overline{y_j^* y_i^*}(A) - \overline{y_j^* y_i^*} \right) \quad j \in V, i \in \bar{j}. \quad (2.64)$$

όπου ο όρος  $\overline{y_j^* y_i^*}(A)$  είναι μια μέση τιμή του  $y_j^* y_i^*$  στη θερμοκή ισορροπία, δεδομένου ότι κρατώνται οι καταστάσεις  $\alpha = (x, d)$  των input και output (ορατών) νευρώνων από το  $A = X \cup Y$ :

$$\overline{y_j^* y_i^*}(A) = \sum_{\alpha \in -1, 1^a} p_d(\alpha) \sum_{\beta \in -1, 1^b} \frac{p_B(\alpha, \beta)}{p_A(\alpha)} y_j^*(\alpha) y_i^*(\alpha) \quad (2.65)$$

όπου  $y_j^*(\alpha)$  είναι ένα output του νευρώνα  $j \in V$  στη θερμοκή ισορροπία της μηχανής Boltzmann δεδομένου ότι οι καταστάσεις των ορατών νευρώνων συνδέονται με ένα διάνυσμα  $\alpha$ . Όμοια, ο όρος  $y_j^* y_i^*$  στην (2.65) είναι μια μέση τιμή του  $y_j^* y_i^*$  στη θερμοκή ισορροπία χωρίς κράτημα των ορατών νευρώνων:

$$\overline{y_j^* y_i^*} = \sum_{y \in -1, 1^s} p_B(y) y_j^* y_i^* \quad (2.66)$$

Αφού αντικατασταθεί η μερική παράγωγος της (2.64) στην (2.63), παίρνουμε τον κανόνα μάθησης της μηχανής Boltzmann:

$$\Delta w_{ji} = \frac{\epsilon}{T^*} \left( \overline{y_j^* y_i^*}(A) - \overline{y_j^* y_i^*} \right) \quad j \in V, i \in \bar{j}. \quad (2.67)$$

Ο πρώτος όρος  $\overline{y_j^* y_i^*}(A)$  αντιστοιχεί στον κανόνα του Hebb με clamping τους input και output (ορατούς) νευρώνες, ενώ ο δεύτερος  $-\overline{y_j^* y_i^*}$  αντιστοιχεί στο Hebbian unlearning με ελεύθερους ορατούς νευρώνες. Η προσαρμοστική φάση συγκλίνει εάν αυτοί οι όροι ταιριάζουν για κάθε  $j \in V, i \in \bar{j}$ .

#### 2.4.5 Αλγόριθμος Μάθησης

Τα παραπάνω συνοψίζονται σε έναν προφανή αλγόριθμο που μπορεί να πραγματοποιηθεί μέσω κάποιου ειδικού hardware [25] ή να προσομοιωθεί σε ένα τυπικό υπολογιστή χρησιμοποιώντας τη μέθοδο Monte Carlo:

1. Αρχικοποίησε το διακριτό χρόνο  $t:=0$ .
2. Επέλεξε τυχαία  $w_{ji}^{(0)} = w_{ij}^{(0)} \in \langle -1, 1 \rangle$  με ομοιόρφη κατανομή πιθανότητας
3. Θέσε  $t := t + 1$ .
4. Ανάθεσε  $(\varrho_{ji}^{\pm})^{(t-1)}$  για κάθε  $j, i$ .
5. Για κάθε πρότυπο εκπαίδευσης  $k = 1, \dots, p$  εκτέλεσε τα παρακάτω  $q$  φορές:
  - (a) Φτιάξε τις καταστάσεις  $\alpha = (x'_k, d'_k) \in \{-1, 1\}^{n+m}$  των ορατών νευρώνων έτσι ώστε η πιθανότητα ότι μια κατάσταση ενός ορατού νευρώνα συμφωνεί με την αντίστοιχη επιθυμητή κατάσταση που περιγράφεται από το  $k$ -οστο πρότυπο είναι ίδια, δηλαδή  $\mathbf{P}\{x'_{k_i} = x_{k_i}\} = 0.9$  ( $i = 1, \dots, n$ ) και  $\mathbf{P}\{d'_{k_j} = d_{k_j}\} = 0.9$  ( $j = 1, \dots, m$ ). Ακόμα, ακολούθησε τις (2.51), (2.52) χρησιμοποιώντας *simulated annealing* έως ότου φτάσεις σε θερμική ισορροπία, στην κατάσταση  $y^*$  με τελική θερμοκρασία  $T^*$ . Η κατάσταση  $y^*$  θα χρησιμοποιηθεί ως αρχική κατάσταση της ακόλουθης μηχανής Boltzmann, με τον υπολογισμό να ξεκινά σε μακροσκοπικό χρόνο  $\tau = 0$ .
  - (b) Σε μακροσκοπικό χρόνο υπολογισμού  $\tau = 1, \dots, r$  κάνε τα ακόλουθα:
    - i. Κάνε ένα μακροσκοπικό βήμα υπολογισμού της μηχανής Boltzmann υπό θερμοκρασία  $T^*$ , δηλαδή με τυχαία σειρά αναβάθμισε όλες τις καταστάσεις των κρυμμένων νευρώνων σύμφωνα με τις (2.51), (2.52).
    - ii. Αναβάθμισε τις στατιστικές για  $\overline{y_j^* y_i^*}(A)$  σύμφωνα με την τρέχουσα κατάσταση του δικτύου, δηλαδή
 
$$(\varrho_{ji}^{\pm})^{(t-1)} = (\varrho_{ji}^{\pm})^{(t-1)} + y_j^{(\tau)} y_i^{(\tau)} .$$
6. Εξακρίβωσε το μέσο του  $(\varrho_{ji}^{\pm})^{(t-1)} := \frac{(\varrho_{ji}^{\pm})^{(t-1)}}{pqr}$   $\{(\varrho_{ji}^{\pm})^{(t-1)} \text{ είναι μια εκτίμηση του τρέχοντος } \overline{y_j^* y_i^*}(A) \}$ .
7. Ανάθεσε  $(\varrho_{ji}^{\pm})^{(t-1)} := 0$  για κάθε  $j, i$ .

8. Κάνε τα παρακάτω  $pq$  φορές:

(a) Για μια τυχαία αρχική κατάσταση της μηχανής Boltzmann ακολούθησε την υπολογιστική δυναμική (2.51), (2.52) (αναβαθμίζονται όλοι οι ορατοί και οι κρυμμένοι νευρώνες) με χρήση *simulated annealing* έως τη θερμική ισορροπία, στην κατάσταση  $y^*$  με τελική θερμοκρασία  $T^*$ . Η κατάσταση  $y^*$  θα χρησιμοποιηθεί ως αρχική κατάσταση της ακόλουθης μηχανής Boltzmann και ο υπολογισμός αρχίζει σε μακροσκοπικό χρόνο  $\tau = 0$ .

(b) Σε μακροσκοπικό χρόνο υπολογισμού  $\tau = 1, \dots, r$  κάνε τα ακόλουθα:

i. Κάνε ένα μακροσκοπικό βήμα υπολογισμού της μηχανής υπό θερμοκρασία  $T^*$ , για παράδειγμα αναβάθμισε με τυχαία σειρά όλες τις καταστάσεις των νευρώνων σύμφωνα με τις (2.51), (2.52).

ii. Αναβάθμισε τις στατιστικές για  $\overline{y_j^* y_i^*}$  σύμφωνα με την τρέχουσα κατάσταση του δικτύου, δηλαδή  
 $(\varrho_{ji}^-)^{(t-1)} = (\varrho_{ji}^-)^{(t-1)} + y_j^{(\tau)} y_i^{(\tau)}$ .

9. Εξακρίβωσε το μέσο του  $(\varrho_{ji}^-)^{(t-1)} := \frac{(\varrho_{ji}^-)^{(t-1)}}{pqr}$   
 $\{(\varrho_{ji}^-)^{(t-1)} \text{ είναι μια εκτίμηση του τρέχοντος } \overline{y_j^* y_i^*}\}$ .

10. Σύμφωνα με την (2.67) υπολόγισε  $\Delta w_{ji}^t = \frac{\epsilon}{T^*} \left( (\varrho_{ji}^+)^{(t-1)} - (\varrho_{ji}^-)^{(t-1)} \right)$ .

11. Σύμφωνα με την (2.62) αναβάθμισε τη διαμόρφωση  $w_{ji}^{(t)} := w_{ji}^{(t-1)} + \Delta w_{ji}^{(t)}$ .

12. Αν  $\Delta w_{ji}^{(t)}$  είναι επαρκώς μικρό, τέλος, διαφορετικά συνέχισε με το βήμα 3.



## Κεφάλαιο 3

# CELLULAR NEURAL NETWORKS

### Εισαγωγή

*Το φαινόμενο της ανάπτυξης των CNN, από τη σύλληψη της βασικής τους αρχιτεκτονικής το 1987 ως την επινόηση της καθολικής CNN μηχανής το 1992, επιβεβαιώνει την ανθεκτικότητά τους στο χρόνο. Σπάνια μια ιδέα έχει προκαλέσει τόσο μεγάλο ενδιαφέρον στους επιστημονικούς κύκλους, έχει ηλεκτρίσει τόσες ερευνητικές δραστηριότητες και έχει δημιουργήσει τόσες πολλές εφαρμογές σε τόσο σύντομο χρονικό διάστημα.*

*Το αρχικό CNN προσφέρει ένα συμβατό με VLSI νευρωνικό δίκτυο αποτελούμενο από τοπικά συνδεδεμένους "νευρώνες". Όλοι οι νευρώνες και οι συνδέσεις τους είναι πανομοιότυποι, για αυτό και κάθε CNN καθορίζεται μοναδικά από 19 το πολύ πραγματικούς αριθμούς (πολλοί εκ των οποίων στην πράξη είναι μηδενικοί), ανεξάρτητα από το μέγεθός του. Με άλλα λόγια, η λειτουργία ενός CNN είτε πρόκειται για πίνακα διάστασης 100 είτε για πίνακα 1.000.000 εικονοστοιχείων, μπορεί να ρυθμιστεί μεταβάλλοντας το πολύ 19 παραμέτρους από το ίδιο κύκλωμα.*

*Από την άποψη του λογισμικού, οι 19 αυτοί αριθμοί μπορούν να ερμηνευτούν ως πρόγραμμα που οδηγεί τους νευρώνες να εκτελέσουν μια συγκεκριμένη εργασία. Αν εφαρμόσουμε τον ίδιο αλγόριθμο σε συμβα-*

τικό υπολογιστή θα γράφαμε ένα πρόγραμμα αποτελούμενο από χιλιάδες εντολές. Με  $19!$  αντιμεταθέσεις σε αυτές τις 19 παραμέτρους, εκ των οποίων ένας αντιστοιχεί σε συγκεκριμένο δίκτυο, είναι ουσιαστική η διαφορά με έναν Von Neumann υπολογιστή. Αλλά είναι ακόμα πιο χρήσιμο αν σκεφτούμε την κάθε αντιμετάθεση των 19 αυτών παραμέτρων σαν αναλογικό πρόγραμμα που δε χρησιμοποιεί τη δύναμη ενός υπολογιστή για να φέρει εις πέρας πετυχημένα αποτελέσματα μαθηματικών υπολογισμών σε δυαδική μορφή, αλλά νόμους της φύσης-όπως τους κανόνες του Kirchhoff-προκειμένου να εφαρμόσει μια παράλληλη δυναμική διαδικασία σε πολύ μικρό πραγματικό χρόνο.

Από την άποψη των φυσικών εξαρτημάτων του υπολογιστή, χρειάζονται μόνο 19 καλώδια από την πλακέτα. Αυτό το χαρακτηριστικό κάνει απλό το σχεδιασμό ενός ολοκληρωμένου κυκλώματος του οποίου η λειτουργία μπορεί να προγραμματιστεί εξωτερικά. Επιπλέον σε ένα συμβατικό υπολογιστή ο υπολογιστικός χρόνος επεξεργασίας μιας εικόνας αυξάνεται εκθετικά σε σχέση με τον πίνακα, ενώ σε ένα CNN αυξάνεται γραμμικά στις περισσότερες περιπτώσεις.

Το αρχικό CNN δίκτυο έχει γενικευτεί από πολλούς ερευνητές προς αρκετές κατευθύνσεις. Η εισαγωγή των μη γραμμικών και *time delay templates* έχει μεγενθύνει το πεδίο δράσης των CNN, περιλαμβάνοντας μεγάλη ποικιλία εφαρμογών όπως ολική βελτιστοποίηση, αρίθμηση, ταξινόμηση, υπολογισμό ιδιοτιμών κλπ. Εξάλλου, μια διακριτού χρόνου εκδοχή ενός CNN που μπορεί εύκολα να πραγματοποιηθεί με *switched capacitor* τεχνολογία, έχει περιγραφεί σε μορφή πλακέτας και έχει να επιδείξει πλήθος χρήσιμων χαρακτηριστικών.

Αλλά η πιο σημαντική γενίκευση των CNN είναι η πρόσφατη εφεύρεση της καθολικής μηχανής (CNN Universal machine) που ενσωματώνει σε ένα κελί, ένα σχετικά μικρό τμήμα της πλακέτας για μια τοπική αναλογική μνήμη, μια τοπική λογική μνήμη και μια μονάδα επικοινωνίας και ελέγχου για τοπική και ολική αλληλεπίδραση. Έτσι έχουμε ένα πρόγραμμα που κληρονομεί τα χαρακτηριστικά ενός υπολογιστή, εκτός από το ότι η μηχανή CNN δουλεύει σε πραγματικό χρόνο. Έχουν λοιπόν δημιουργηθεί CNN γλώσσα και CNN λειτουργικό σύστημα τα οποία επιτρέπουν στο χρήστη να γράψει ένα πρόγραμμα για επεξεργασία εικόνας μέσω μιας ακολουθίας από CNN με διαφορετικά templates, καθένα για διαφορετική δουλειά. Σε αντίθεση με τον κλασσικό υπολογιστή όπου κάθε λέξη στον καταχωρητή είναι μια αλληλουχία από δυαδικούς



αριθμούς, κάθε λέξη στη λογική μνήμη περιέχει μόνο μια κατάσταση (πχ. 'YES' ή 'NO').

Η καθολική CNN μηχανή, που συνδυάζει αναλογικό και λογικό κύκλωμα (δυναμική πραγματικού χρόνου και λογικές συναρτήσεις AND, OR, NOR κλπ) είναι το μόνο γνωστό νευρωνικό δίκτυο του οποίου η δυναμική μοιάζει με τις λειτουργίες αυτές του εγκεφάλου που δείχνουν σαν αναλογικές και εκτελούνται στο δεξί ημισφαίριο, αλλά και με τις αντίστοιχες "λογικές" που εκτελούνται στο αριστερό. Αναπαριστά δηλαδή αρκετά καλά τη λεγόμενη "εγκεφαλική ασυμμετρία" του ανθρώπινου εγκεφάλου.

Πρόσφατα αποδείχτηκε ότι η καθολική μηχανή CNN είναι μια αναλογική μηχανή Turing. Συγκεκριμένα, μια αναλογική εκδοχή του παιχνιδιού της Ζωής εφαρμόστηκε σαν μια μηχανή CNN, αποδεικνύοντας την καθολικότητά της υπό την έννοια του Turing. Αυτό το θεμελιώδες αποτέλεσμα αποδεικνύει ότι η μηχανή CNN μπορεί να λύσει οποιοδήποτε πρόβλημα λύνει ένας απλός υπολογιστής.

Είναι λοιπόν αρκετά ώριμη περίοδος για μεταφορά τεχνολογιών και εφαρμογών από άλλες περιοχές, καθώς η άφιξη της μηχανής υποστηρίζεται από πολλούς αλγόριθμους μάθησης και προσαρμοστικότητας. Αυτές οι τεχνολογίες μπορεί να περιλαμβάνουν ηλεκτρομαγνητικές, οπτικές εφαρμογές ακόμα και εφαρμογές χβαντομηχανικής. Αυτό το κεφάλαιο καλύπτει τέσσερις περιοχές τους: θεωρία, VLSI εφαρμογές, CNN εργαλεία σχεδιασμού και εφαρμογές.

## 3.1 THE CNN PARADIGM

### 3.1.1 Εισαγωγή

Η σύλληψη των CNN έγινε το 1987 στο εργαστήριο του καθηγητή Leon O. Chua στο Berkeley. Η παρουσίαση των L. O. Chua και L. Yang (*IEEE Transactions on Circuits and Systems*, Οκτώβριος 1988) έπεισε πολλούς ερευνητές ότι πρόκειται για ένα πολύ ισχυρό νέο παράδειγμα και μέσα σε δύο μόλις χρόνια πάνω από εκατό συγγραφείς έστειλαν δημοσιεύσεις στο πρώτο Διεθνές IEEE Εργαστήριο για τα CNN και τις εφαρμογές τους. Αυτή η συνάντηση στη Βουδαπέστη ήταν ιδιαίτερα επιτυχημένη και σύντομα το "International Journal of Circuit Theory and Applications" ανακοίνωσε ότι θα αφιερωθεί ένα ειδικό τεύχος στο θέμα. Από την εφεύρεσή τους διατήρησαν δύο βασικά χαρακτηριστικά: γεωμετρικά τοπική συνδετικότητα του τρισδιάστατου πίνακα του επεξεργαστή και την αναλογική δυναμική των κελιών και των αλληλεπιδράσεών τους. Έχουν αναπτυχθεί τόσο ώστε να καλύπτουν μια πολύ μεγάλη κλάση προβλημάτων και πλαισίων εργασίας, και η τελευταία ανακάλυψη της καθολικής CNN μηχανής παρέχει μια εννιαία πλατφόρμα και μια πρωτότυπη αρχιτεκτονική για ένα νέο τύπο αποθηκευτικού προγράμματος: τον αναλογικό<sup>1</sup> αλγόριθμο CNN. Οι πολλές εφαρμογές του παραδείγματος (επεξεργασία εικόνας, ανάλυση προτύπων συμπεριλαμβανομένου σχήματος, κίνησης και χρώματος, επίλυση μερικών διαφορικών εξισώσεων, σχεδιασμός διάφορων βιολογικών εφαρμογών κλπ) που εμφανίστηκαν σε λίγα μόλις χρόνια από την εφεύρεσή τους, και οι ακριβείς αποδεδειγμένες θεωρητικές τους ικανότητες δικαιολογούν το ασυνήθιστο ενδιαφέρον στην περιοχή αυτή. Τα πρώτα δοκιμασμένα CNN chips υπόσχονται πρωτοφανή υπολογιστική δύναμη, της τάξης των *tera XPS* ( $10^{12}$ ) πράξεις ανά δευτερόλεπτο σε ένα chip  $2\text{ cm}^2$ .

---

<sup>1</sup>Με την έννοια "αναλογικός", θεωρούμε τα δύο κύρια χαρακτηριστικά του chip των CNN, "αναλογικό" και "λογικό" υπολογισμό, κατά τα οποία διαφέρει από άλλους παράλληλους υπολογιστές. Δίνουμε έμφαση στο ότι η "λογική" περιέχει μόνο τις συμβολικές μεταβλητές "yes" και "no". Δεν υπάρχει δυαδική A/D μετατροπή κατά τη διάρκεια του υπολογισμού.

### 3.1.2 Το απλό πλαίσιο λειτουργίας του CNN

Στην απλούστερη περίπτωση, το CNN είναι ένας πίνακας απλών, πανομοιότυπων, μη γραμμικών, δυναμικών κυκλωμάτων τοποθετημένων σε διδιάστατο πλέγμα. Αν δε σημειώνεται διαφορετικά, μιλάμε για ορθογώνιο πλέγμα. Μεταφέροντας τα πλέγματα στις τρεις διαστάσεις, έχουμε *multi-layered CNN*. Κάθε κόμβος στο πλέγμα καταλαμβάνεται από ένα κελί, η δομή του φαίνεται στο Σχήμα 3.1. Ο πυκνωτής  $C$  και η αντίσταση  $R_x$  αποτελούν τον πυρήνα του κελιού. Η ηλεκτρική τάση  $u_{xij}$  στα άκρα του πυκνωτή στον κόμβο  $(i, j)$  ελέγχει τη μη γραμμική πηγή στην έξοδο του κελιού που παράγει τάση  $u_{yij}$ . Συνήθως αυτή η πηγή έχει σιγμοειδή με ξεχωριστά βήματα γραμμική χαρακτηριστική  $f(\cdot)$ , που φαίνεται στο Σχήμα 3.2. Στην πραγματικότητα, αυτά τα στοιχεία αναπαριστούν έναν απλό πολλαπλασιαστή με είσοδο τον πυκνωτή και την αντίσταση. Κάθε κελί δέχεται εξωτερικά σήματα μέσω του input  $u_{xij}$  και έχει μια ανεξάρτητη πηγή  $I_{ij}$ . Η τάση ενός δοσμένου κελιού δεν επηρεάζεται μόνο από τα δικά του input και, μέσω ανάδρασης, output, αλλά και από τις input και output τάσεις των γειτονικών κελιών. Έστω οι κανονικοποιημένες παράμετροι ( $C = 1, R_x = 1, R_y = 1$ ) και οι γραμμικές αλληλεπιδράσεις, η δυναμική του πίνακα με  $M \times N$  κελιών, υπολογίζεται από τις ακόλουθες εξισώσεις:

$$\begin{aligned} \dot{u}_{xij}(t) = & -u_{xij}(t) + \sum_{C(kl) \in N_r(ij)} A(ij; kl)u_{ykl} \\ & + \sum_{C(kl) \in N_r(ij)} B(ij; kl)u_{ukl} + I_{ij} \end{aligned} \quad (3.1)$$

$$u_{yij} = f(u_{xij}) \quad (3.2)$$

$1 \leq i \leq M$   $1 \leq j \leq N$  όπου  $u_{xij}(t)$ ,  $u_{xij}(t)$  και  $u_{yij}(t)$  είναι τα input, state και output σήματα του κελιού  $C_{ij}$ , αντίστοιχα. Κάθε κελί  $C_{ij}$  αλληλεπιδρά με τα γειτονικά του ( $C_{kl}$ ) σε μια ορισμένη περιοχή (γειτονιά)  $N_r(ij)$  που ορίζεται ως εξής

$$C_{kl} \in N_r(ij) \Leftrightarrow \max\{|k - i|, |l - j|\} \leq r \quad (3.3)$$

Τα κελιά που αντιστοιχούν σε γειτονίες ακτίνας  $r = 1$  σε ορθογώνιο πλέγμα και αυτά που αντιστοιχούν σε ακτίνα  $r = 2$  σε εξάγωνο πλέγμα φαίνονται στα σχήματα 3.1a και 3.1c, αντίστοιχα.

Σχήμα 3.1: Διαφορετικοί τύποι πλεγμάτων. Τα κελιά με τη βούλα δείχνουν τις γειτονιές των μαύρων κελιών για  $r = 1$  και  $r = 2$  σε διαφορετικά πλέγματα.

Σχήμα 3.2: Ένα απλό CNN κελί με μη-γραμμικότητα

Όπως γίνεται ξεκάθαρο από τη σχέση (3.1), δοθέντος της αρχικής κατάστασης και ενός input σήματος, η συμπεριφορά του συστήματος καθορίζεται πλήρως από τους συντελεστές αλληλεπίδρασης  $A(ij;kl)$  και  $B(ij;kl)$ . Ένας πίνακας CNN αυθαίρετου μεγέθους καθορίζεται από λίγες μόνο παραμέτρους. Αυτό το σύνολο παραμέτρων, καθορισμένο σε πίνακες, καλείται *cloning template*. Στην περίπτωση της μικρότερης περιοχής ( $r = 1$ ):

$$A = \begin{bmatrix} a_{i-1,j-1} & a_{i,j-1} & a_{i+1,j-1} \\ a_{i-1,j} & a_{i,j} & a_{i+1,j} \\ a_{i+1,j+1} & a_{i,j+1} & a_{i+1,j+1} \end{bmatrix}, B = \begin{bmatrix} b_{i-1,j-1} & b_{i,j-1} & b_{i+1,j-1} \\ b_{i-1,j} & b_{i,j} & b_{i+1,j} \\ b_{i+1,j+1} & b_{i,j+1} & b_{i+1,j+1} \end{bmatrix}, I = c$$

είναι αντίστοιχα τα *feedback template*, *forward template* και *bias term* που αποτελούν ένα *cloning template*.

Οι έμφυτες ικανότητες του CNN μπορούν να επεκταθούν εισάγοντας μη γραμμικές ( $\hat{A}_{ij,kl}$ ) και *delay type*  $A_{ij,kl}^T$  αλληλεπιδράσεις. Η εξίσωση της κατάστασης (*state equation*) τροποποιείται:

$$\begin{aligned} \dot{u}_{xij}(t) = & - u_{xij}(t) + I \\ & + \sum_{C(kl) \in N_r(ij)} \hat{A}_{ij,kl}(u_{ykl}, u_{yij}) + \sum_{C(kl) \in N_r(ij)} \hat{B}_{ij,kl}(u_{ukl}, u_{xij}) \\ & + \sum_{C(kl) \in N_r(ij)} A_{ij,kl}^T u_{ykl}(t) + \sum_{C(kl) \in N_r(ij)} B_{ij,kl}^T u_{ukl}(t) \end{aligned} \quad (3.4)$$

Ένα σημαντικό χαρακτηριστικό των CNN είναι ότι έχει δύο ανεξάρτητες ικανότητες: *generic input* και την αρχική κατάσταση των κελιών. Συνήθως φράσσονται από

$$|u_{xij}(t)| \leq 1 \text{ και } |u_{xij}(0)| \leq 1 \quad (3.5)$$

Όμοια, αν  $|f(\cdot)| \leq 1$  τότε  $|u_{yij}(t)| \leq 1$ .

Όταν χρησιμοποιείται για επεξεργασία πινάκων, το CNN κάνει την αντιστοίχιση

$$u_{xij}(0)u_{xij}(t) \mapsto u_{yij}(t) \quad (3.6)$$

μέσω συνάρτησης  $F$  του *cloning template*  $T(A, B, I)$ . Για παράδειγμα, μια ασπρόμαυρη εικόνα μπορεί να κωδικοποιηθεί έτσι ώστε το +1 να αντιστοιχεί στα μαύρα εικονοστοιχεία (*pixels*) και το -1 στα λευκά, και

Σχήμα 3.3: Τα input και output ενός υπολογισμού

αν αυτή η εικόνα φορτωθεί με αρχικές καταστάσεις  $u_{xij}(0)$  στα κελιά, τότε η αντιστοιχία  $F(T)$  είναι ένας επεξεργαστής εικόνας με output  $u_{yij}(\infty)$ . Ορίζουμε ως *cloning template*  $T$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 0,$$

τότε το  $u_{yij}(t)$  τείνει στις σταθερές τιμές  $[-1, 1]$  και το CNN υπολογίζει το μέσο όρο σε ένα  $3 \times 3$  παράθυρο. Παράδειγμα βλέπουμε στην εικόνα (3.3).

### 3.1.3 Ο γενικός ορισμός του CNN παραδείγματος

Ορισμός 3.1.1 Το CNN είναι

- 2-, 3- ή  $n$ -διαστάσεων πίνακας από
- πανομοιότυπα δυναμικά συστήματα, που καλούνται κελιά και ικανοποιούν δυο ιδιότητες:
- οι περισσότερες αλληλεπιδράσεις μεταξύ των κελιών είναι τοπικές και μέσα σε μια ακτίνα  $r$  και
- όλες οι μεταβλητές είναι *continuously valued functions*.

**Παρατήρηση 3.1.1** 1. Ο χώρος των μεταβλητών είναι πάντα διακριτοποιημένος: κάθε κελί ορίζεται από 2, 3 ή  $n$  ακέραιους  $(i, j, k, \dots, n)$ .

2. Η μεταβλητή του χρόνου  $t$  μπορεί να είναι συνεχής ή διακριτή.

3. Η αλληλεπίδραση που παρουσιάζεται από το *cloning template* μπορεί να είναι μια μη γραμμική συνάρτηση της κατάστασης  $x$ , του *output*  $y$  και του *input*  $u$  του κάθε κελιού, μέσα στη γειτονιά  $N_r$ , ακτίνας  $r$ , όπως και του χρόνου  $t$ . Το *cloning template* μπορεί να είναι ανεξάρτητο ή όχι του χώρου.

4. Το δυναμικό σύστημα καθορίζεται μοναδικά από κάποιον κανόνα (ODEs, συναρτησιακές εξισώσεις, εξισώσεις διαφορών κλπ) ώστε τα δοθέντα  $x(t)$  και  $u(t)$  για κάθε  $t \in (-\infty, t_0)$  ή  $t \in (-\infty, \dots - \Delta t, 0, \Delta t, \dots, n_0 \Delta t)$ ,  $x(t)$  και  $u(t)$  είναι μοναδικά ορισμένα για κάθε  $t \geq t_0$ , ή  $n \Delta t$  με  $n \geq n_0$ . Περιλαμβάνει και τις συνοριακές συνθήκες.

5. Συνήθως, το δυναμικό σύστημα και/ή οι αλληλεπιδράσεις μπορούν να διαταραχθούν με θόρυβο.



### 3.1.4 Παραδείγματα Εφαρμογών

**Εφαρμογή 3.1.1** Μια μη γραμμική φόρμα για υπολογισμό ολικού μεγίστου

Όταν εφαρμοστεί σε τοπολογική ανάλυση εικόνας, το CNN προσφέρει ένα ευέλικτο και ικανό εργαλείο για υπολογισμό γεωμετρικών σχημάτων. Αυτό το παράδειγμα δείχνει πως η αξία και η ακριβής θέση του μεγίστου μιας αυθαίρετης τριδιάστατης επιφάνειας, μπορεί να καθοριστεί με δυο φόρμες (templates). Η βασική ιδέα είναι η εξής: θεωρούμε ένα CNN πλέγμα με αρχικές συνθήκες ( $u_{xij}(0)$ ) και κανονικοποιημένες τιμές στο  $[-1, +1]$ . Η συμπεριφορά των κελιών πρέπει να υπακούει τους ακόλουθους κανόνες:

1. Αν ένα κελί έχει μεγαλύτερο output από τα γειτονικά του, η τιμή της κατάστασης (και του output) του κελιού παραμένει σταθερή.
2. Αν υπάρχουν μεγαλύτερα outputs στη γειτονιά του κελιού, η τιμή της κατάστασης (και του output) αρχίζει να αυξάνεται εως ότου γίνει ίση με τη μεγαλύτερη της γειτονιάς.

Έτσι όλα τα outputs θα γίνουν ίσα με το μεγαλύτερο της γειτονιάς, το ολικό μέγιστο. Το δεύτερο βήμα έχει να κάνει με την ανίχνευση της ακριβούς θέσης του μεγίστου, χρησιμοποιώντας την ακόλουθη απλή φόρμα:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I = 0.05,$$

Αυτή η φόρμα συγκρίνει το αποτέλεσμα μεταξύ της πρώτης και της αρχικής εικόνας που φορτώθηκαν στα inputs και στις αρχικές καταστάσεις του πίνακα, αντίστοιχα.

**Εφαρμογή 3.1.2** Ένα four-layer CNN για έγχρωμη φωτοσιγχογραφία

Φωτοσιγχογραφία είναι η διαδικασία με την οποία μια ασπρόμαυρη φωτογραφία γεννιέται από μια grey-scale, έτσι ώστε όταν παρατηρήσουμε την ασπρόμαυρη από κάποια κατάλληλη απόσταση, μοιάζει όμοια με την grey-scale.

Σχήμα 3.4: Υπολογισμός ολικού μεγίστου: η αρχική επιφάνεια (a), η τιμή (b), και η θέση (c) του.

Η διαδικασία αυτή επιτυγχάνεται με μια δομή CNN που περιέχει τέσσερα επίπεδα. Η αρχιτεκτονική φαίνεται στο σχήμα 3.5. Το input και η αρχική κατάσταση των τριών πρώτων επιπέδων είναι οι τρεις αναλογικές μονόχρωμες φωτογραφίες, και μια φωτογραφία με οκτώ χρώματα θα εμφανιστεί στο output του τέταρτου επιπέδου.

Οι φόρμες του multi-layer CNN είναι οι εξής:

$$A_{11} = A_{22} = A_{33} = \begin{bmatrix} -0.03 & -0.08 & -0.13 & -0.08 & -0.03 \\ -0.08 & -0.36 & -0.6 & -0.36 & -0.08 \\ -0.13 & -0.6 & 1.25 & -0.6 & -0.13 \\ -0.08 & -0.36 & -0.6 & -0.36 & -0.08 \\ -0.03 & -0.08 & -0.13 & -0.08 & -0.03 \end{bmatrix}$$

$$B_{11} = B_{22} = B_{33} = \begin{bmatrix} 0 & 0 & 0.33 & 0 & 0 \\ 0 & 0.24 & 0.47 & 0.240 & 0 \\ 0.33 & 0.47 & 1.45 & 0.47 & 0.33 \\ 0 & 0.24 & 0.47 & 0.240 & 0 \\ 0 & 0 & 0.33 & 0 & 0 \end{bmatrix}$$

$$I_1 = I_2 = I_3 = I_4 = 0$$

$$A_{41} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.125 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A_{42} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A_{43} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Η έγχρωμη φωτοσιγγογραφία γίνεται στα πρώτα τρία επίπεδα. Στο τέταρτο επίπεδο, αναπαράγουμε μια φωτογραφία οκτώ χρωμάτων. Στο σχήμα 3.6 βλέπουμε την αρχική φωτογραφία και την αναπαράστασή της με έγχρωμη φωτοσιγγογραφία. Λόγω του δυαδικού output των τριών πρώτων επιπέδων, τα κελιά του τέταρτου επιπέδου θα έχουν τις ακόλουθες τιμές:

$$(\pm 1) \times 0.125 + (\pm 1) \times 0.25 + (\pm 1) \times 0.5 =$$

Σχήμα 3.5: Η δομή του multi-layer CNN που χρησιμοποιείται για φωτοσιγχογραφία

Σχήμα 3.6: Η αρχική εικόνα (a) και η αναπαράστασή της (b)

$\{-0.875; -0.625; -0.375; -0.125; -0.125; -0.375; -0.625; -0.825\}$

*Η κωδικοποίηση των χρωμάτων είναι η εξής:*

λευκό (κόκκινο + πράσινο + μπλε):  $-0.875$

τυρκουάζ (πράσινο + μπλε):  $-0.625$

λιλά (κόκκινο + μπλε):  $-0.375$

μπλε (μπλε):  $-0.125$

κίτρινο (κόκκινο + πράσινο):  $0.125$

πράσινο (πράσινο):  $0.375$

κόκκινο (κόκκινο):  $0.625$

μαύρο (χωρίς χρώματα):  $0.875$

*Βλέποντας την κωδικοποίηση των χρωμάτων και τους πίνακες A που συνδέουν το τέταρτο επίπεδο, παρατηρούμε πχ, το μπλε χρώμα εμφανίζεται σε ένα pixel μόνο αν το αντίστοιχο pixel στο τρίτο επίπεδο (το μπλε επίπεδο) είναι  $\pm 1$ . Αν και το αποτέλεσμα περιέχει οκτώ μόνο χρώ-*

*ματα, από μια σχετική απόσταση, η φωτογραφία μοιάζει με την αρχική  
έγχρωμη, με πολλές σκιές και αποχρώσεις.*

## 3.2 CNNs ΔΙΑΚΡΙΤΟΥ ΧΡΟΝΟΥ (Discrete-Time CNNs)

### 3.2.1 Εισαγωγή

Η αρχιτεκτονική των cnn είναι πολύ δυνατή στην αναγνώριση προτύπων και στην επεξεργασία εικόνας. Η απλή δομή ενός κελιού και η συνδεκτικότητα τους κάνουν τα cnn ικανά για εφαρμογή σε αναλογικά VLSI. Το δίκτυο περιγράφεται από ένα σύστημα μη γραμμικών ΔΕ

$$\frac{dx^c}{dt} = -x^c(t) + \sum_{d \in N_r(c)} a_d^c y^d + \sum_{d \in N_r(c)} b_d^c u^d + i^c \quad (3.7)$$

$$y^c(t) = \frac{1}{2} (|x^c(t) + 1| - |x^c(t) - 1|) \quad (3.8)$$

Η μεταβλητή  $x^c$  δηλώνει την κατάσταση του κελιού  $c$ , η  $y^c$  το output και η  $u^c$  το input. Η κατάσταση ενός κελιού ελέγχεται από τα inputs και outputs των παρακείμενων κελιών εντός της  $r$ -γειτονιάς  $N_r(c)$ . Τα output πολλαπλασιάζονται με το συντελεστή  $a_d^c$  και αναδρούν, ενώ τα inputs πολλαπλασιάζονται με τις παραμέτρους ελέγχου  $b_d^c u^d$ . Το σύνολο των αναδράσεων, συντελεστών ελέγχου και κατωφλιών καλείται template. Η  $r$ -γειτονιά  $N_r(c)$  είναι το σύνολο όλων των κελιών απόστασης  $r$  από το κελί  $c$ . Υπάρχουν προβλήματα σχετικά με το σύνορο του πλέγματος, επειδή δεν υπάρχουν όλα τα γειτονικά κελιά. Έτσι εισάγονται τεχνητά κελιά με σταθερές τιμές για τα  $y^c$  και  $u^c$ . Στη συνέχεια, οι τιμές  $u^c = -1$  και  $y^c = -1$  χρησιμοποιούνται για αυτά τα κελιά, που σημαίνει ότι ο input και ο αρχικός πίνακας περιβάλλονται από ένα δακτύλιο λευκών εικονοστοιχείων. Μια  $r$ -γειτονιά μπορεί να οριστεί σε ένα εξαγωνικό πλέγμα.

Το output  $y^c$  λαμβάνεται από μια κομματιαστή (piecewise) γραμμική συνάρτηση της κατάστασης του κελιού στη σχέση (3.8). Ανάλογα με την αρχική κατάσταση  $x(t_0)$ , το σύστημα συγκλίνει σε μια τελική κατάσταση  $x(t_\infty)$  με δυαδικά outputs αν οι συντελεστές ανάδρασης είναι συμμετρικοί και η αυτο-ανάδραση (self-feedback) είναι αρκετά μεγάλη ( $a_c^c \geq 1$ ).

Ο δείκτης  $d$  εφαρμόζεται πάντα μέσα στην  $r$ -γειτονιά ενός κελιού  $c$ , όπου ο δείκτης  $c$  χρησιμοποιείται για όλα τα κελιά του πλέγματος.

Έτσι η (3.7) μπορεί να ξαναγραφτεί ως

$$\frac{dx^c}{dt} = -x^c(t) + a_d^c y^d(t) + b_d^c u^d + i^c \quad (3.9)$$

### 3.2.2 Αρχιτεκτονική

Τα CNN διακριτού χρόνου (DTCNNs) είναι μια ειδική περίπτωση δικτύου με αναδράσεις όπου οι τοπικές αλληλοσυνδέσεις και τα σταθερά βάρη μεταφέρονται από τα CNN. Περιγράφονται πλήρως από έναν περιοδικά επαναλαμβανόμενο αλγόριθμο. Η δυναμική συμπεριφορά οφείλεται στην ανάδραση των δυαδικών outputs. Ένα κελί επηρεάζεται από τα inputs και outputs των γειτονικών κελιών μέσα σε μια  $r$  - γειτονιά. Η αρχιτεκτονική μοιάζει με αυτή των cellular automata αλλά διαφέρει στο ότι έχει inputs και βάρη με συνεχείς τιμές.

#### Ορισμός 3.2.1 CNNs Διακριτού χρόνου

Τα DTCNNs ορίζονται από τον ακόλουθο αλγόριθμο χρησιμοποιώντας το δείκτη  $d$  μέσα στην  $r$  - γειτονιά του κελιού  $c$ :

$$x^c(k) = a_d^c y^d + b_d^c u^d + i^c \quad (3.10)$$

$$y^c(k) = f(x^c(k-1)) = \begin{cases} 1, & \text{if } x^c(k-1) > 0 \\ -1, & \text{if } x^c(k-1) < 0. \end{cases} \quad (3.11)$$

Σε αντίθεση με τα CNNs, μόνο δυαδικές τιμές (-1, +1) δίνονται σαν βάρη των συντελεστών  $a_d^c$  του τελεστή ανάδρασης. Πριν αρχίσει η επανάληψη, οι αρχικές τιμές  $y^c(0) \in \{-1, 1\}$  πρέπει να οριστούν. Είναι κρίσιμες για τη δυναμική συμπεριφορά του συστήματος και μπορούν να θεωρηθούν σαν ένα δεύτερο input του κελιού. Η τιμή +1 δηλώνει ένα μαύρο εικονοστοιχείο και -1 ένα λευκό. Τα inputs πολλαπλασιάζονται με τις παραμέτρους ελέγχου  $b_d^c$  για όλα τα γειτονικά κελιά  $d$  μέσα στην  $N_r(c)$ . Μια σταθερή τιμή  $i^c$  προστίθεται για να ρυθμίσει το κατώφλι του κελιού  $c$ . Μια τιμή  $i$  χωρίς δείκτη σημαίνει ότι όλα τα κελιά έχουν την ίδια τιμή. Συνεχείς συντελεστές και inputs στα templates θεωρούνται σταθεροί του χρόνου, αλλά μια επέκταση σε παραμέτρους εξαρτημένες



από το χρόνο, είναι πιθανή. Η δυαδική κατάσταση του output  $y^c$  καθορίζεται από το πρόσημο του  $x^c$  της προηγούμενης επανάληψης. Δεν ορίζεται όταν  $x^c = 0$ . Στην πράξη υπάρχει πάντα θόρυβος, που σημαίνει ότι  $|x^c| > 0$ .

Αν οι συντελεστές των templates επιλεγθούν έτσι ώστε

$$\Delta = \min_{c,k} |a_d^c y^d(k) + b_d^c u^d + i^c| \quad (3.12)$$

να είναι αρκετά μεγάλο, ο αλγόριθμος είναι σχετικά απαθής στην ανοχή (tolerance) των  $a_d^c, b_d^c$  ή  $i^c$  μικρότερο του  $\Delta$ , επειδή δεν μπορούν να προκαλέσουν ιδιαίτερες αλλαγές στις καταστάσεις των outputs.

Συνοψίζοντας, το κομμάτι της (3.10) που είναι ανεξάρτητο χρόνου, ορίζεται ως το bias του κελιού  $k^c(u)$ , που είναι πάλι συνάρτηση των σταθερών inputs και των τιμών των κατωφλιών:

$$k^c = k^c(u) = b_d^c u^d + i^c \quad (3.13)$$

Το πλεονέκτημα των DTCNNs είναι η περιγραφή της επόμενης κατάστασης του output μέσω γραμμικών ανισοτήτων. Η προσομοίωση σε ψηφιακό υπολογιστή είναι πολύ απλή, αφού το σύστημα είναι διακριτού χρόνου και δεν είναι απαραίτητος κάποιος πολύπλοκος αλγόριθμος. Στο σχήμα (3.7) δίνεται η δομή του δικτύου για τον αλγόριθμο (3.10), (3.11). Οι μεταβλητές αντικαθιστούνται από τάσεις και τα βήματα των επαναλήψεων από στιγμιότυπα διακριτού χρόνου.

Το επόμενο επαναληπτικό βήμα λαμβάνεται μετά από μια περίοδο των σιναλών του ρολογιού  $\phi_1$  και  $\phi_2$  (Σχήμα 3.8).

Οι πολλαπλασιασμοί γίνονται από **linear voltage-controlled current sources**. Το συνολικό ρεύμα μεταφέρεται σε μια τάση  $v_x^c(kT)$  από την αντίσταση  $R_x$  μέσω της

$$v_x^c(kT) = R_x (A_d^c v_y^d(kT) + B_d^c v_u^d + I^c) \quad (3.14)$$

Οι **voltage-control** ηλεκτρικές πηγές  $F(v_x^c(kT))$  και  $F(v_z^c(kT))$  έχουν τη μη γραμμική χαρακτηριστική

$$F(v) = \begin{cases} V_{sat} & \text{αν } v > 0 \\ -V_{sat} & \text{αν } v < 0 \end{cases} \quad (3.15)$$

Εδώ ορίζεται η επόμενη κατάσταση του output και ο πυκνωτής  $C_1$  φορτίζεται κατά την 'υψηλή' φάση του  $\phi_1$  σε  $V_{sat}$  ή  $-V_{sat}$ . Κατά τη

Σχήμα 3.7: Δικτυακή δομή ενός απλού κελιού

Σχήμα 3.8: Διάγραμμα CNNs διακριτού χρόνου

διάρκεια της 'υψηλής' φάσης του  $\phi_2$  ( $\phi_1$  "χαμηλό") οι πυκνωτές  $C_1$  και  $C_2$  συνδέονται παράλληλα. Η τάση  $v_z^c(kT)$  ρυθμίζεται μετά την αιφνίδια μεταβολή τάσης, σε

$$v_z^c(kT) = \frac{C_1 F(v_x^c((k-1)T)) + C_2 v_z^c((k-1)T)}{C_1 + C_2} \quad (3.16)$$

---

Πίνακας 1. Κανονικοποίηση των μεταβλητών  
και των στοιχείων του δικτύου

---

$$\begin{array}{ll}
 u^c = v_u^c/V_{sat} & a_d^c = A_d^c R_x \\
 x^c(k) = v_x^c(kT)/V_{sat} & b_d^c = B_d^c R \\
 z^c(k) = v_z^c(kT)/V_{sat} & i^c = I^c r_x/V_{sat} \\
 y_k^c = v_y^c(kT)/V_{sat} &
 \end{array}$$


---

Επειδή  $|F(v_x^c)| = V_{sat}$  και  $|v_z^c| \leq V_{sat} \forall k$ , προκύπτει ότι για  $C_1 > C_2$

$$v_y^c(kT) = F(v_x^c((k-1)T)) = \begin{cases} V_{sat} & \text{αν } v_x^c((k-1)T) > 0 \\ -V_{sat} & \text{αν } v_x^c((k-1)T) < 0 \end{cases} \quad (3.17)$$

Εισάγοντας την (3.14) στην (3.17) έχουμε

$$v_y^c(kT) = F(R_x(A_d^c v_y^d((k-1)T) + B_d^c v_u^d + I^c)) \quad (3.18)$$

Εφορμόζοντας την κανονικοποίηση στον Πίνακα 1, προκύπτει ο αλγόριθμος των (3.10), (3.11).

Συγκρίνοντας με τα CNNs, η αναλογική αντίληψη των DTCNNs έχει σημαντικά πλεονεκτήματα.

- Λόγω των δυαδικών outputs, η αλληλεπίδραση αρκετών κυκλωμάτων είναι πολύ απλή. Για τα CNNs, τα συνεχή output σήματα πρέπει να πολλαπλασιαστούν (propagated) κατά τη διάρκεια της μεταβολής της τάσης.
- Η ανοχή των παραμέτρων της (3.12) κάνει το δίκτυο πιο δυνατό στην κατασκευή αν τα templates σχεδιαστούν κατάλληλα.
- Η ταχύτητα της διάδοσης propagation μπορεί να ελεγχθεί μόνο μεταβάλλοντας το ρυθμό του ρολογιού. Αυτό απλουστεύει επίσης την ικανότητα δοκιμής ενός chip.

**Παρατήρηση 3.2.1** Ένα DTCNN που περιέχει μια αρκετά μεγάλη γειτονιά μπορεί να θεωρηθεί σαν ένα perceptron με ένα επίπεδο με μια συσκευή σύγκρισης μη γραμμικότητας και αναδράσεων ή σαν ένα διακριτό Hopfield δίκτυο που αναβαθμίζεται παράλληλα. Όμως οι μεγάλες

γειτονίες δεν είναι κατάλληλες για εφαρμογή σε απλά VLSI. Δίνονται δυνατοί περιορισμοί για τους συντελεστές των βαρών. Αν καταργηθεί η σταθερότητα, η δομή του δικτύου μπορεί να χρησιμοποιηθεί ως συσχετιζόμενη μνήμη, εκπαιδευμένη με τον κανόνα του Hebb.

### 3.2.3 Εφαρμογές

Οι περισσότερες εφαρμογές των CNNs μπορούν να αποδοθούν και από τα DTCCNs. Μία καινοτομία στην επεξεργασία εικόνας είναι η μεταφορά αντικειμένων στις ομόκεντρους ισοϋψείς τους και η αναζήτηση αντικειμένων με την ελάχιστη απόσταση από σταθερό σημείο. Για τα CNNs δεν έχει βρεθεί ακόμα φόρμα για κάτι τέτοιο.

#### Ορισμός 3.2.2 Four-connected

Δυο εικονοστοιχεία  $(i, j)$  και  $(m, n)$  του ίδιου χρώματος είναι *four-connected* αν

$$|i - m| + |j - n| \leq 1 \quad (3.19)$$

#### Ορισμός 3.2.3 Eight-connected

Δυο εικονοστοιχεία  $(i, j)$  και  $(m, n)$  του ίδιου χρώματος είναι *eight-connected* αν

$$|i - m| \leq 1 \wedge |j - n| \leq 1 \quad (3.20)$$

#### Ορισμός 3.2.4 Four-connected path

Ένα *four-connected* μονοπάτι  $P$  ορίζεται σαν ένα σύνολο από  $r$  εικονοστοιχεία όπου δύο διαδοχικά εικονοστοιχεία  $p_i$  και  $p_{i+1}$  είναι *four-connected*.

#### Ορισμός 3.2.5 Four-connected object

Ως *four-connected object*  $O$  ορίζουμε το μέγιστο σύνολο πανομοιότυπων έγχρωμων εικονοστοιχείων όπου για κάθε δύο από αυτά  $(i, j), (m, n) \in O$  υπάρχει ένα *four-connected* μονοπάτι  $P \subseteq O$ .

**Παρατήρηση 3.2.2** Η 1 - γειτονιά ορίστηκε για *eight-connected* εικονοστοιχεία στον Ορισμό 3.2.3. Μια *four-connected* γειτονιά αποτελείται από κελιά που επηρεάζονται μόνο από *four-connected* εικονοστοιχεία. Τα *four-connected* εικονοστοιχεία είναι πάντα *eight-connected*. Για μονοδιάστατα προβλήματα ή για ένα εξαγωνικό πλέγμα, αυτή η διάκριση είναι περιττή.

### Εφαρμογή 3.2.1 Linear thresholding

Περιγραφή προβλήματος Μπορεί να θεωρηθεί σαν συνέλιξη. Οι συντελεστές ανάρδρασης ρυθμίζονται ίσοι με μηδέν και το πρότυπο εφαρμόζεται στα input των κελιών. Οι συντελεστές ελέγχου χρησιμοποιούνται για να εξάγουν συγκεκριμένα χαρακτηριστικά του input προτύπου μέσα στην  $r$ - γειτονιά κάθε κελιού. Αυτό αντιστοιχεί στο bias  $k^c$  του κελιού που εκτελεί μια διχρητή συνέλιξη με έναν πυρήνα  $\mathbf{b}$ :

$$x^c = k^c = b_a^c u^d + i^c \quad (3.21)$$

Η τιμή του  $i^c$  ρυθμίζει ένα κατώφλι προκειμένου να δεχτεί την απόφαση του εντοπισμένου χαρακτηριστικού.

Απόδειξη σύγκλισης Ένα DTCNN που χρησιμοποιείται σαν δίκτυο με γραμμικό κατώφλι συγκλίνει μετά από μια επανάληψη για κάθε παράμετρο  $b_a^c$ ,  $u^d$  και  $i^c$ , επειδή το δίκτυο δεν έχει ανάρδραση. Η ακόλουθη κατάσταση του output εξαρτάται μόνο από τα inputs.

Ακολουθεί παράδειγμα με μαύρα eight-connected objects. Αυτό μπορεί να επιλυθεί με χρήση του template

$$a = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 \cdot 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad i = -1 \cdot 5 \quad (3.22)$$

Η συνθήκη προκειμένου να εξασφαλιστεί ένα μαύρο output κελιού  $i, j$  είναι

$$4 \cdot 5 u^{i,j} - u^{i-1,j} - u^{i,j-1} - u^{i,j+1} - u^{i+1,j} - 1 \cdot 5 > 0 \quad (3.23)$$

πχ. ένα κελί μένει μαύρο μόνο όταν υπάρχει τουλάχιστον ένα λευκό γειτονικό κελί (ορισμός 3.2.2). Μπορεί να αποδειχτεί ότι οι άκρες του προτύπου με θόρυβο αναγνωρίζονται σωστά αν περιοριστούν στο σύνολο ορισμού  $D = \{u^c \mid 0 \cdot 89 < |u^c| < 1 \cdot 11\}$ .

### Εφαρμογή 3.2.2 Hole filler

Περιγραφή προβλήματος Τα άσπρα εικονοστοιχεία που περικλείονται

σε ένα μαύρο *eight-connected object*  $O$  πρέπει να είναι μαύρα στο τελικό *output*.

Αυτό μπορεί να επιλυθεί με διάδοση που αρχίζει στα σύνορα του πλέγματος. Η αρχική κατάσταση είναι μαύρη (+1). Το πρότυπο χρησιμοποιείται σαν *input* των κελιών. Μεταβάσεις από μαύρο σε λευκό στο *output* επιτρέπονται μόνο αν το *input* του κελιού είναι λευκό και υπάρχει ένα *four-connected* λευκό γειτονικό κελί στη γειτονιά. Αυτό ικανοποιείται για

$$y^{i,j}(k+1) = -1 \Rightarrow u^{i,j} = -1 \wedge y^{i-1,j}(k) + y^{i,j-1}(k) + y^{i,j+1}(k) + y^{i+1,j}(k) < +4 \quad (3.24)$$

$$y^{i,j}(k+1) = +1 \Rightarrow u^{i,j} = +1 \vee y^{i-1,j}(k) + y^{i,j-1}(k) + y^{i,j+1}(k) + y^{i+1,j}(k) = +4 \quad (3.25)$$

και αυτό μπορεί να γίνει χρησιμοποιώντας το *template*

$$a = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad b = \boxed{0 \mid 4 \cdot 5 \mid 0}, \quad i = -1 \quad (3.26)$$

Απόδειξη σύγκλισης Ένα *DTCNN* που χρησιμοποιεί το *template* (3.26) συγκλίνει λόγω του Θεωρήματος 1 αν τα *inputs* υπόκεινται στον περιορισμό

$$D = \{u^c \mid |u^c| > \frac{2}{3}\} \quad (3.27)$$

Πρόκειται για χαρακτηριστική συνάρτηση (Ορισμός 3.2 1) επειδή  $k_{min}^c > 2$  και η συνθήκη

$$\min_c(3 + k_{min}^c) > 4$$

ικανοποιείται για όλους τους συνδυασμούς του *input* προτύπου.

### Εφαρμογή 3.2.3 Ομόκεντρα Σχήματα

Περιγραφή προβλήματος Ως ομόκεντρο σχήμα ενός μαύρου **eight-connected object**  $O$  ορίζεται σαν ένα σύνολο εναλασσόμενων μαύρων

και άσπρων δακτυλίων που ξεκινούν από το σύνορο προς το εσωτερικό. Στο σχ.(3.9) βλέπουμε παράδειγμα.

Αυτό μπορεί να χρησιμοποιηθεί για ένα είδος αυτο - αναγνώρισης αντικειμένων. Αντικείμενα διαφόρων μεγεθών, αλλά με παρόμοιους ομόκεντρους δακτυλίους μπορούν να εντοπιστούν με *linear thresholding*.

Ένα *DTCNN* το επιλύει εξάγοντας τον εξωτερικό δακτύλιο και αντιστρέφοντας τα εσωτερικά εικονοστοιχεία του. Ο ισοϋψής δακτύλιος παραμένει σταθερός κατά τις διαδοχικές επαναλήψεις. Ένας νέος ομόκεντρος δακτύλιος δημιουργείται σε κάθε επανάληψη. Οι δακτύλιοι που προκύπτουν μπορούν να θεωρηθούν σαν μια διάδοση προς το εσωτερικό του αντικειμένου. Ένα *output*  $y^{i,j}$  επιτρέπεται να αλλάξει κατάσταση αν υπάρχουν τέσσερα γειτονικά κελιά  $y^{i-1,j}$ ,  $y^{i,j-1}$ ,  $y^{i,j+1}$ , και  $y^{i+1,j}$  (λόγω του Ορισμού 5) του ίδιου χρώματος. Εφαρμόζοντας το πρότυπο στην αρχική κατάσταση  $y(0)$  και στα *inputs* του κελιού  $\mathbf{u}$ , χρησιμοποιούμε το *template*

$$a = \begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 3 \cdot 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}, \quad b = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 4 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \quad i = -4 \quad (3.28)$$

Σχήμα 3.9: Τετράγωνο και το ομόκεντρό του σχήμα

Επιλέγοντας  $b_{i,j}^{i,j}$  και  $i = -4$  βεβαιωνόμαστε ότι το  $x^{i,j}$  δεν μπορεί να είναι θετικό αν  $u^{i,j} = -1$ . Επιπλέον, πρέπει να τηρούνται οι εξής προϋποθέσεις:

$$u^{i,j} = +1 \wedge y^{i,j}(k) = y^{i-1,j}(k) = y^{i,j-1}(k) = y^{i,j+1}(k) = y^{i+1,j}(k) = +1 \Rightarrow x^{i,j} < 0 \quad (3.29)$$

$$u^{i,j} = +1 \wedge y^{i,j}(k) = y^{i-1,j}(k) = y^{i,j-1}(k) = y^{i,j+1}(k) = y^{i+1,j}(k) = -1 \Rightarrow x^{i,j} > 0 \quad (3.30)$$

Απόδειξη σύγκλισης Ένα *DTCNN* με το *template* (3.28) συγκλίνει για όλα τα *inputs*. Σε κάθε επανάληψη ο αριθμός των κελιών που επεξεργάζονται τα τέσσερα γειτονικά μειώνεται, γιατί ο εξαγόμενος δακτύλιος πάντα γίνεται σταθερός. Σε λιγότερες από  $n/2$  επαναλήψεις δεν υπάρχει κελί με τέσσερα ίδια γειτονικά.

**Παρατήρηση 3.2.3** Για "ανάλυση" ενός αντικειμένου οποιουδήποτε μεγέθους στους ομόκεντρους δακτύλιους του, είναι απαραίτητο μόνο ένα *template* γειτονιάς  $r = 1$ . Για αυτήν την εφαρμογή δεν έχει βρεθεί ακόμα *template* για *CNNs*.

**Εφαρμογή 3.2.4** Αύξηση και μείωση αντικειμένων βήμα προς βήμα Περιγραφή προβλήματος Ένα μαύρο αντικείμενο  $O$  μπορεί να μεγαλώσει ή να μικρύνει σε κάθε επανάληψη κατά ένα επίπεδο εικονοστοιχείων.

Το *DTCNN* πρέπει να προγραμματιστεί σαν **cellular automaton**. Το πρότυπο χρησιμοποιείται σαν αρχική κατάσταση. Όσο μεγαλώνει το αντικείμενο, επιτρέπεται μόνο μια μετάβαση της *output* κατάστασης από λευκό σε μαύρο αν το κελί έχει τουλάχιστον ένα μαύρο γείτονα. Έτσι, το αντικείμενο εμπλουτίζεται σε κάθε επανάληψη με έναν καινούριο δακτύλιο από μαύρα εικονοστοιχεία.



Το πρόβλημα επιλύεται χρησιμοποιώντας το *template*

$$a = \begin{array}{|c|c|c|} \hline 0.5 & 0.5 & 0.5 \\ \hline 0.5 & 0.5 & 0.5 \\ \hline 0.5 & 0.5 & 0.5 \\ \hline \end{array}, \quad b = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \quad i = +4 \quad (3.31)$$

επειδή

$$\exists d \in N_r(c) : y^d(k) = +1 \Rightarrow y^c(k+1) = +1 \quad (3.32)$$

Απόδειξη σύγκλισης Ένα *DTCNN* που χρησιμοποιεί αυτό το *template* συγκλίνει λόγω του Θεωρήματος 1. Το *template* είναι **eigendominant** λόγω του Ορισμού 4 επειδή  $k_{min}^c = 4$  και η συνθήκη

$$\min_c(a_{(c)}^{(c)} + k_{min}^c) = 4 \cdot 5 > 4$$

ικανοποιείται για όλους τους συνδυασμούς του αρχικού προτύπου.

Για να μικρύνει ένα μαύρο αντικείμενο  $O$ , μόνο μεταβάσεις από μαύρο σε λευκό επιτρέπονται αν το κελί έχει τουλάχιστον ένα λευκό γειτονικό. Επομένως το αντικείμενο μικραίνει σε κάθε επανάληψη κατά ένα δακτύλιο μαύρων εικονοστοιχείων. Αυτό γίνεται μεταβάλλοντας το  $i$  στην (3.31) σε  $-4$ .

**Παρατήρηση 3.2.4** Το *template* (3.31) αυξάνει αντικείμενο που αντιστοιχεί σε μια **eight-connectivity**. Αυτό μπορεί να γίνει και για **four-connectivity** αν χρησιμοποιηθεί το *template*

$$a = \begin{array}{|c|c|c|} \hline 0 & 0.5 & 0 \\ \hline 0.5 & 0.5 & 0.5 \\ \hline 0 & 0.5 & 0 \\ \hline \end{array}, \quad b = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \quad i = +2 \quad (3.33)$$

## 3.3 CNNs WITH NON-LINEAR AND DELAY-TYPE TEMPLATE ELEMENTS AND NON-UNIFORM GRIDS

### 3.3.1 Εισαγωγή

Το CNN παράδειγμα αποτελεί μια δυνατή μη-γραμμική υπολογιστική δομή για ποικιλία υπολογισμών σε πίνακες, συνήθως τοποθετημένους σε ισόπλευρο πλέγμα. Εισάγουμε τώρα αρχιτεκτονικές με μη-ομοιόμορφους επεξεργαστές και γειτονιές (πλέγματα). Οι υπολογισμοί των πινάκων μπορούν να θεωρηθούν ως παράλληλη εκτέλεση πολύπλοκων λειτουργιών σε ένα μεγάλο αριθμό επεξεργαστών τοποθετημένων σε ένα γεωμετρικά ομοιόμορφο πλέγμα. Αν πρόκειται για λογικές πράξεις, το κλασικό **cellular automaton (CA)** του John von Neumann και οι πρόσφατες παραλλαγές του, είναι τα ιδανικά εργαλεία. Αν οι πράξεις είναι αριθμητικές (8, 16 ή 32 bits) οι **systolic** πίνακες είναι η καλύτερη προσέγγιση. Και στις δυο περιπτώσεις χρησιμοποιούμε ψηφιακούς επεξεργαστές. Αν όμως οι τιμές των σημάτων είναι συνεχείς, και/ή πραγματοποιούνται *real time* εφαρμογές, τα CNNs είναι η ιδανική λύση. Η απλότητα της δομής τους, η μεγάλη τους σταθερότητα και η μεταβλητότητα της λειτουργίας τους είναι ιδανικά για εφαρμογή σε VLSI.

Διατηρώντας τα χαρακτηριστικά των CNNs και εισάγοντας πολύ απλά μη-γραμμικά και *delay-type templates*, το CNN μπορεί να γίνει ένα δυνατό πλαίσιο εργασίας για αναλογική δυναμική πινάκων. Από θεωρητική άποψη, ένα CNN έχει καλά ορισμένες ποιοτικές ιδιότητες και περιέχει ως ειδικές περιπτώσεις μια μεγάλη ποικιλία δυνατών λύσεων για επεξεργασία προτύπων. Από πρακτική άποψη, το CNN παρέχει ένα φυσικό παράδειγμα για σχεδιασμό αναλογικών VLSI που μπορούν να προγραμματιστούν ή οπτικοακουστικών *real-time* πινάκων υπολογισμού.

### 3.3.2 Το γενικό πλαίσιο λειτουργίας του CNN με μη-γραμμικά και *delay-type templates*

Ας θεωρήσουμε ένα πλέγμα δυο διαστάσεων όπως στο σχήμα (3.10) Τα τετράγωνα αναπαριστούν τους επεξεργαστές και οι συνδέσεις τους τις πιθανές γραμμές επικοινωνίας με τους γείτονές τους. Το διδιάστατο

αυτό πλέγμα επεξεργαστών καλείται **layer**.

Ο αναλογικός επεξεργαστής, το κελί, φαίνεται στο σχήμα (3.11). Διαφέρει από το κλασικό κελί κυρίως στις πηγές  $I_{xy}$  και  $I_{xu}$ . Δηλαδή αντί για τις δυο γραμμικά ελεγχόμενες πηγές που ορίζονται από

$$A(ij;kl)u_{ykl}$$

και

$$B(ij;kl)u_{ukl}$$

που συνδέονται με ένα τυπικό κελί  $C_{ij}$  και ένα τυπικό κελί  $C_{kl}$  με το οποίο αλληλεπιδρά, επιτρέπουμε μη - γραμμικές και **delayed controlled** πηγές

$$\hat{A}_{ij;kl}(u_{ykl}, u_{yij}) + A_{ij;kl}^T(u_{ykl}(t - \tau))$$

και

$$\hat{B}_{ij;kl}(u_{ukl}, u_{uij}) + B_{ij;kl}^T u_{ukl}(t - \tau)$$

όπου τα  $u$ ,  $x$ , και  $y$  δηλώνουν μεταβλητές input, κατάστασης και output αντίστοιχα. Είναι πιθανό να ισχύει  $\tau = \tau_{kl}$ .

Αυτό σημαίνει ότι αντί για μια γραμμική VCCS (voltage-controlled current source) στα  $A$  και  $B$  templates, έχουμε μια μη-γραμμική και/ή delay-type VCCS. Η δομή της μη-γραμμικότητας στα templates είναι επίσης σημαντική: είναι μια συνάρτηση με δύο το πολύ μεταβλητές, την output τάση του κελιού  $C_{ij}$  και την αντίστοιχη του γειτονικού του  $C_{kl}$ .

Επιτρέπουμε επίσης στην output συνάρτηση να έχει μια μεγαλύτερη εμβέλεια output τάσεων, έτσι ώστε η **saturation** τάση να είναι  $\pm K$  αντί για  $\pm 1$ . Αυτές οι πολύ μικρές αλλαγές βελτιώνουν την κλάση της δυναμικής των CNNs αισθητά.

Ορίζουμε λοιπόν την  $r$  - γειτονιά του κελιού  $C_{ij}$  με  $M \times N$  κελιά ως

$$N_r(ij) = \{C_{kl} : \max(|k - i|, |l - j|) = r(\text{integer})\} \quad (3.34)$$

Ας διατυπώσουμε τώρα τις κανονικές εξισώσεις περιγραφής του αναλογικού μη-γραμμικού πίνακα επεξεργασίας του CNN με μη-γραμμικά και delay-type templates. Χωρίς περιορισμό της γενικότητας, υποθέτουμε ότι τα κελιά του επεξεργαστή σχεδιάζονται σε ορθογώνιο πλέγμα

Σχήμα 3.10: Τυπικά κανονικά πλέγματα: ορθογώνιο (τέσσερις ή οκτώ γείτονες), τριγωνικό (έξι γείτονες), εξαγωνικό (τρεις γείτονες)

Σχήμα 3.11: Το κύκλωμα του αναλογικού επεξεργαστή του κελιού και η μη-γραμμική output συνάρτηση

με  $M \times N$  κελιά,  $1 < i < M$ ,  $1 < j < N$ . Αυτό είναι ένα **delay CNN** ή ένα μη-γραμμικό CNN. Ομοίως, όταν έχουμε τη δυνατότητα να έχουμε περισσότερους από έναν τύπο επεξεργαστών και/ή περισσότερες από μια διαστάσεις σε μια γειτονιά ενός CNN, μπορούμε να αποκαλούμε αυτό το CNN σαν μη-ομοιόμορφο επεξεργαστή CNN ή σαν **multiple-neighbourhood-size CNN**, αντίστοιχα.

Αναφερόμενοι στα σχήματα (3.10) και (3.11), μπορούμε να ορίσουμε ένα CNN μαθηματικά ως εξής:

### Εξίσωση κατάστασης

$$\begin{aligned}
C\dot{u}_{xij} &= -(1/R_x)u_{xij}(t) + I + \sum_{C_{kl} \in N_r(ij)} \hat{A}_{ij;kl}(u_{ykl}(t))(u_{yij}(t)) \\
&+ \sum_{C_{kl} \in N_r(ij)} \hat{B}_{ij;kl}(u_{ukl}(t), u_{yij}(t)) + \sum_{C_{kl} \in N_r(ij)} A_{ij;kl}^T u_{ykl}(t - \tau) \\
&+ \sum_{C_{kl} \in N_r(ij)} B_{ij;kl}^T u_{ukl}(t - \tau) \tag{3.35}
\end{aligned}$$

όπου  $\hat{A}$ ,  $\hat{B}$  και  $A^T$ ,  $B^T$  συνδέονται με τα μη γραμμικά και **delay-type cloning templates** αντίστοιχα. Συγκεκριμένα,  $\hat{A}_{ij;kl}$  και  $\hat{B}_{ij;kl}$  είναι συνεχείς συναρτήσεις το πολύ δυο μεταβλητών και  $A_{ij;kl}^T$  και  $B_{ij;kl}^T$  είναι πραγματικές σταθερές. Αν γράψουμε  $A(ij;kl)$  και  $B(ij;kl)$ , εννοούμε ένα απλό γραμμικό template.

Για παράδειγμα, με  $d_1 = c_1 [\exp(u_{ykl}) - 1]$  και  $d_2 = c_2 [\exp(u_{ykl}) - u_{yij}]$  μπορούμε να ορίσουμε συμβολικά ένα cloning template

$$\hat{A} = \begin{bmatrix} 0 & d_1 & 0 \\ d_1 & 2 & d_1 \\ 0 & d_1 & 0 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 & 0 & 0 \\ d_2 & 1 & d_2 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 0$$

### Output εξίσωση

*H output* συνάρτηση

$$u_{yij}(t) = f(u_{xij}(t)) \tag{3.36}$$

αναπαριστάται στο σχήμα (3β). Είναι μια συνάρτηση *piecewise* γραμμική με συνεχή κλίση στο διάστημα  $(-K, K)$ ,  $K > 0$ . Μπορεί να προσεγγιστεί με κάθε ακρίβεια από μια ομαλή ( $C^1$ ) αυστηρά αυξανούσα σιγμοειδή συνάρτηση.

Γενικεύοντας, μπορούμε να επιτρέψουμε στην *output* συνάρτηση να έχει τη δική της δυναμική. Για παράδειγμα, η εξίσωση (3.36) μπορεί να αντικατασταθεί από μια πρώτης τάξης εξίσωση κατάστασης

$$\dot{u}_{yij} = -\alpha u_{yij} + f(u_{xij}(t)) \quad (3.37)$$

ή από μια υψηλότερης τάξης.

Όμως, για να παραμείνει αύξουσα η μονοτονία της  $f$  θα πρέπει να γίνει πιο πολύπλοκη.

### Input εξίσωση

$$u_{uij} = E_{ij} \quad (3.38)$$

### Εξισώσεις περιορισμών

$$|u_{xij}(0)| \leq 1 \quad (3.39)$$

$$|u_{uij}(t)| \leq 1 \quad (3.40)$$

### Υποθέσεις παραμέτρων

$$K \geq 0, \quad C \geq 0, \quad R_x \geq 0, \quad \tau \geq 0 \quad (3.41)$$

Για την περίπτωση που  $\hat{A}_{ij;kl}$  και  $\hat{B}_{ij;kl}$  είναι γραμμικές συναρτήσεις και  $\tau = 0$ , ή αντίστοιχα όταν

$$\hat{A}_{ij;kl} = A(ij;kl)u_{ykl}, \quad \hat{B}_{ij;kl} = B(ij;kl)u_{ukl}$$

και οι  $A_{ij;kl}^T, B_{ij;kl}^T$  είναι μηδενικοί, έχουμε το αρχικό CNN με γραμμικά **cloning templates**.

Υποκινούμενοι κυρίως από νευρο-βιολογικές δομές, αξίζει να αναφέρουμε τις αρχιτεκτονικές CNN με μη-ομοιόμορφο πλέγμα, που έχουν

περισσότερους από έναν τύπο επεξεργαστή και/ή γειτονίες με διαφορετικές διαστάσεις. Θα αναφερόμαστε σε ένα τέτοιο CNM ως NUP - CNN (non-uniform processor CNN) ή ως MNS - CNN (multiple-neighbourhood-size CNN), αντίστοιχα.

Στο σχήμα βλέπουμε ένα NUP - CNN με δυο τύπους επεξεργαστών, οι οποίοι συμβολίζονται από τα λευκά και τα μαύρα τετράγωνα. Μια ειδική περίπτωση αυτού του διπλού CNN (συμπεριλαμβανομένων και των λογικών επεξεργαστών) είναι η CNNL αρχιτεκτονική. Πρόκειται για μια NUP - CNN αρχιτεκτονική, η οποία περιέχει περισσότερους από έναν τύπο επεξεργαστές, τοποθετημένους σε ομοιόμορφο πλέγμα και οι δομές των αλληλεπιδράσεων είναι ανεξάρτητες του χώρου.

Στο σχήμα (3.12c) παρατηρούμε ένα MNS - CNN με γειτονίες δύο διαφορετικών μεγεθών. Οι επεξεργαστές είναι όλοι ίδιοι. Όμως, το πλέγμα A είναι ένα πλέγμα με γειτονιά μεγέθους  $r = 1$ , ενώ το πλέγμα B στο επόμενο επίπεδο (διακεκομμένη γραμμή) αφορά τους επεξεργαστές με γειτονιά μεγέθους  $r = 3$ . Στο επίπεδο B εμφανίζονται συνδέσεις μόνο για έναν επεξεργαστή. Τα MNS - CNN περιέχουν επίπεδα με διαφορετικά πλέγματα και διαστάσεις γειτονιών. Μια ειδική περίπτωση του MNS - CNN με δύο διαστάσεις γειτονιών περιέχει μόνο έναν επεξεργαστή στο B επίπεδο.

Λόγω της επεξεργασίας ή της συνάρτησης υπολογισμού του πίνακα του CNN, έχει δύο μέσα εισόδου και ένα μέσο εξόδου για κάθε κελί, επίσης το input  $E_{ij}(t) = u_{xij}(t)$  είναι η αρχική τιμή της κατάστασης, δηλαδή  $u_{xij} = 0$ , μπορεί επίσης να χρησιμοποιηθεί ως input. Το output  $u_{yij}(t)$  ορίζεται σαν η DC σταθερή κατάσταση  $u_{yij}(\infty)$  ή σαν στιγμιότυπο σε μια δεδομένη στιγμή  $t = T$ , πχ  $u_{yij}(T)$ . Είναι ξεκάθαρο ότι η πιο άμεση περιοχή εφαρμογών των CNN είναι η επεξεργασία προτύπων. Παρ'όλα αυτά, άλλες εφαρμογές όπως η επίλυση ειδικών τύπων μερικών διαφορικών εξισώσεων είναι επίσης πολύ σημαντικές.

**Πρόταση 3.3.1** Για ένα μη-γραμμικό CNN που χαρακτηρίζεται από φραγμένα μη-γραμμικά cloning templates (αλλά χωρίς delay) και μια piecewise γραμμική output εξίσωση (εξίσωση (3.36b)) όλες οι καταστάσεις  $u_{xij}$  είναι φραγμένες για κάθε  $t > 0$  και το φράγμα  $u_{max}$  υπολογίζεται από τη φόρμουλα



Σχήμα 3.12: Μη ομοιόμορφοι επεξεργαστές και μεγέθη πλεγμάτων: (a) μη ομοιόμορφος επεξεργαστής CNN, (b) μια ειδική περίπτωση, (c) γειτονιά CNN με διαφορετικές καταστάσεις)

$$u_{max} = 1 + R_x |I| + R_x \max \left( K \sum_{C_{kl} \in N_r(ij)} \max |\hat{A}_{ij;kl}| + \sum_{C_{kl} \in N_r(ij)} \max |\hat{B}_{ij;kl}| \right) \quad (3.42)$$

όπου  $1 \leq i \leq M$ ,  $1 \leq j \leq N$

### 3.3.3 Ειδικές Περιπτώσεις

Όπως ήδη αναφέρθηκε, ένα CNN με μη-γραμμικά και delay τύπου cloning templates μπορεί να θεωρηθεί σαν αντιπροσωπευτικό παράδειγμα για αναλογικούς μη γραμμικούς επεξεργαστές, τοποθετημένους σε κανονικό τρισδιάστατο πλέγμα. Ένα επίπεδο αυτής της παράταξης τοποθετείται σε διδιάστατο κανονικό πλέγμα (βλ. σχήμα (3.10)). Θα περιγράψουμε μια σημαντική ειδική περίπτωση.

Σχήμα 3.13: Η δομή του κυκλώματος ενός ωμικού πλέγματος. Οι αντιστάσεις είναι όλες μη-γραμμικές και χαρακτηρίζονται από  $i = G(u)$

**Μια ειδική περίπτωση: μη-γραμμικό ωμικό πλέγμα (με πυκνωτές)**

Αρκετοί πρόσφατοι επεξεργαστές που μιμούνται απλά μοντέλα του σπονδυλωτού αμφιβληστροειδούς χιτώνα του ματιού ή κάποιο λειτουργικό μοντέλο της οπτικής επεξεργασίας, χρησιμοποιούν τη δομή του κυκλώματος του σχήματος (3.13).

Δηλαδή, επιλέγοντας

$$I = 0, \quad A^T = B^T = 0$$

Επιλέγουμε  $I_{xu} = I_b = B_{ij;ij}(E_{ij})$  έτσι ώστε

$$\hat{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & g & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Επιλέγοντας  $u_{yij} = f(u_{xij})$  με  $K \gg 1$  γι' αυτό  $u_{yij} = u_{xij}$ ,  $R_x = 0$ ,  $\hat{A}_{ij;kl} = G(u)$  ή  $0$  και  $G = G(u_{ykl} - u_{yij})$ , έτσι ώστε

$$\hat{A} = \begin{bmatrix} 0 & G & 0 \\ G & 0 & G \\ 0 & G & 0 \end{bmatrix}$$

Σχήμα 3.14: Μια input εικόνα

Τονίζουμε ότι το CNN που ορίστηκε μπορεί να λειτουργήσει σε δύο καταστάσεις, είτε στη μεταβατική κατάσταση, είτε στην DC σταθερής κατάστασης. Στη μεταβατική το output είναι το στιγμιότυπο σε μια δεδομένη στιγμή  $t = T$ . Στην DC σταθερής κατάστασης το output είναι η σταθερή κατάσταση του κυκλώματος.

**Παρατήρηση 3.3.1** Οι παραπάνω περιπτώσεις είναι όντως ειδικές. Σαν παράδειγμα θεωρούμε το input του σχήματος (3.14). Αυτή η φωτογραφία εφαρμόζεται σαν αρχική κατάσταση  $u_{xij}(0)$  σε ένα CNN με τα ακόλουθα γραμμικά templates σε ένα τετράγωνο πλέγμα:

$$\hat{A} = A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{B} = 0, \quad I = 0$$

Το output  $u_{yij}(\infty)$  (σχήμα 3.15a) δίνει τον αριθμό των συνδεδεμένων συνιστωσών όταν η εικόνα (3.14) προβληθεί κατά τον οριζόντιο άξονα. Αν στη συνέχεια επιλέξουμε

$$\hat{A} = A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{B} = 0, \quad I = 0, \quad A^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B^T = 0$$

και  $\tau = 3$ . Το output του σχήματος (3.15b) δίνει ένα συνδυασμό ενός ανιχνευτή συνδεδεμένων συνιστωσών και ενός ανιχνευτή κάθετης γραμμής με delay. Το νέο CNN επομένως, δίνει ένα ποιοτικά καλό αποτέλεσμα στην επεξεργασία εικόνας.

Σχήμα 3.15: Οι output εικόνες με χρήση (a) γραμμικού cloning template και (b) delay-type cloning template

Ένα παράδειγμα μη-γραμμικού CNN προβάλλεται στο σχήμα (3.16). Το cloning template είναι το ακόλουθο,

$$\hat{B} = \begin{bmatrix} 0 & g & 0 \\ g & 0 & g \\ 0 & g & 0 \end{bmatrix}, \quad \hat{A}_{ij,ij} = 2, \quad \hat{A}_{ij,kl} = 0 \quad (ij \neq kl), \quad I = 0$$

όπου  $g(u_{ukl} - u_{yij})$  έχει σταθερή τιμή 0.5 εκτός από το διάστημα -0.18 με 0.18 όπου έχει τιμή -1. Η input εικόνα είναι μια ασπρόμαυρη εκδοχή τμήματος ενός πίνακα του El Greco και το output δείχνει τις ισοϋψείς.

Οι υπολογισμοί έγιναν χρησιμοποιώντας το πρόγραμμα προσομοίωσης CNND σε ένα IBM PS/2 Model 60 υπολογιστή.

Σχήμα 3.16: (a) Η input εικόνα (τμήμα του πίνακα του El Greco) και  
(b) η output εικόνα χρησιμοποιώντας μη-γραμμικό template

### 3.3.4 Τα CNN ως αναλογικός πίνακας υπολογισμού

Μπορούμε να συγκρίνουμε το *input* και την προδιαγραφή ενός *cellular automaton (CA)*, ενός συμβολικού πίνακα (*SA*) και ενός *CNN* ως εξής:

	CA	SA	CNN
Input	Logic Values (1-4 bits)	Numerical Values (8-32 bits)	Analogue Values
Specification	truth table	Numerical Algorithm	Cloning Templates

Ένα δύσκολο πρόβλημα, ωστόσο, είναι η ικανότητα προγραμματισμού. Τα *CA* και *SA* προγραμματίζονται εύκολα. Η προγραμματιστική ικανότητα των *CNN* δεν είναι προφανής. Αποδεικνύεται όμως, ότι τα *CNN* είναι στην πραγματικότητα ένας ικανός να προγραμματιστεί αναλογικός *VLSI* υπολογιστής.

Τα βασικά **building blocks** (άτομα) του *CNN* είναι οι συντελεστές των *cloning templates*, δηλαδή

$$\hat{A}_{ij;kl}, \hat{B}_{ij;kl}, \hat{A}_{ij;kl}^T, \hat{B}_{ij;kl}^T, I$$

Σημειώνουμε ότι το *I* μπορεί να θεωρηθεί σαν ένα *VCCS* με σταθερή ηλεκτρική τάση.

Αυτά τα αναλογικά *templates* είναι όντως ικανά να προγραμματιστούν, αφού κάθε προσθετός σταθερός όρος μπορεί εύκολα να επιλεγεί αν κάνουμε μεταγωγή με διακόπτη την αντίστοιχη συνιστώσα από ένα σύνολο παράλληλων γραμμικών και μη-γραμμικών πηγών.

Οι σειριακές ή παράλληλες εφαρμογές αυτών των εντολών αποτελούν τον αναλογικό αλγόριθμο ή το αναλογικό λογισμικό του *CNN*.

Analogue CNN	Digital CA or SA
Sequence of templates (series of parallel)/analogue SW	Recursive function/algorithm/SW
Cloning template/Analogue 'macro'	Machine instruction/Macro
Cloning template atom	'Gate' (physical circuit)

Σχήμα 3.17: Σχήματα διάταξης

### 3.3.5 Μερικές Ποιοτικές ιδιότητες

Προκειμένου να εφαρμόσουμε κάποια ποιοτικά αποτελέσματα από τη θεωρία δυναμικών συστημάτων, ανασχηματίζουμε τις κανονικές εξισώσεις περιγραφής της δυναμικής του CNN. Χωρίς περιορισμό της γενικότητας, υποθέτουμε ότι  $R_x = 1$ ,  $C = 1$  και  $M = N$ .

Στην περίπτωση γραμμικών templates έχουμε την απλή φόρμα

$$\dot{v}_x = -Uv_x + A^L v_y + B^L v_u + I \quad (3.43)$$

όπου  $v_x(t), v_y(t), v_u(t) \in R^{N^2} \times R^1$ ,  $I \in R^{N^2}$  με όλα τα στοιχεία ίσα με  $I$ , ο  $U$  ένας  $N^2 \times N^2$  μοναδιαίος πίνακας,  $v_y = f(v_x)$  μια διαγώνια σχεδίαση με  $u_{yi} = f(u_{xi})$  και  $A^L, B^L \in R^{N^2 \times N^2}$ .

Έχουμε  $N^2$  κελιά,  $N^2$  στοιχεία στα διανύσματα. Το ερώτημα είναι πώς τα διατάσσουμε. Στο σχήμα (3.17) βλέπουμε τρεις πιθανότητες για την περίπτωση τετράγωνου πλέγματος με  $N = 5$ .

Οι κοντινότεροι γείτονες στις τρεις διατάξεις του σχήματος μπορούν να βρεθούν στα διανύσματα ως οι ακόλουθες αποστάσεις:

$$N + 1, \quad 2N - 2, \quad 2N - 1 \quad (3.44)$$

Δηλαδή, έχουμε  $13 - 7 = 6$  στο σχήμα (3.17a),  $19 - 11 = 8$  στο σχήμα (3.17b) και  $15 - 6 = 9$  στο σχήμα (3.17c).

Για παράδειγμα, θεωρούμε το γραμμικό cloning template

Σχήμα 3.18:  $A^L$

$$A = \begin{bmatrix} 0 & p_N & 0 \\ p_W & 2 & p_E \\ 0 & p_S & 0 \end{bmatrix}, \quad B = 0, \quad I = 0 \quad (3.45)$$

και επιλέγουμε τη διαγώνια διάταξη του (3.17b). Ο πίνακας  $A^L$  που συνδέεται με αυτήν τη διάταξη, φαίνεται παραπάνω, με μερικές δοσμένες τιμές σε επιλεγμένες γραμμές και στήλες.

Οι επιλεγμένες γραμμές συμπληρώνονται. Παρατηρούμε ότι αν ο  $A$  είναι κεντρικά συμμετρικός, θα είναι και ο  $A^L$ .

Για τη γενική περίπτωση ενός CNN με μη-γραμμικά και delay-type templates και χρησιμοποιούμε τις ίδιες μεταβλητές και διανύσματα, όπως προηγουμένως, η κανονική περιγραφή πίνακα-διανύσματος είναι η ακόλουθη (θεωρούμε πάλι ένα τετράγωνο πλέγμα με  $r = 1, R_x = 1, C = 1$ ):

$$\dot{v}_x = -Uv_x + \hat{A}(v_y)e + \hat{B}(v_u)e + A^T v_y(t - \tau) + B^T v_u(t - \tau) + I \quad (3.46)$$

όπου  $e = [1, 1, 1, \dots, 1]^T \in R^{N^2}$ ,  $\hat{A}$  και  $\hat{B}$  είναι  $(N^2 \times N^2)$  πίνακες με



στοιχεία  $\hat{A}_{ij;kl}(v_{ykl}, v_{yij})$  και  $\hat{B}_{ij;kl}(v_{ukl}, v_{uij})$  αντίστοιχα και  $A^T$  και  $B^T$  ( $N^2 \times N^2$ ) πίνακες με στοιχεία  $A_{ij;kl}^T$  και  $B_{ij;kl}^T$  αντίστοιχα.

## 3.4 VLSI IMPLEMENTATION

### 3.4.1 Εισαγωγή

Η εφαρμογή των VLSI κυκλωμάτων στα αναλογικά CNN (ACNNs), έχει αναφερθεί αρκετά πρόσφατα. Είναι δύσκολος ο έλεγχός τους, αλλά η αποδοτικότητα στην τελική εκτέλεση του συστήματος έχει αναγνωριστεί. Θα προτείνουμε διαδικασίες ελέγχου που εκμεταλλεύονται ακριβώς τη φύση της αρχιτεκτονικής των CNNs.

### 3.4.2 Υποθέσεις

Για λόγους απλότητας, θα αρκεστούμε σε ACNNs με τους ακόλουθους περιορισμούς.

- Πολυπλοκότητα Θεωρούμε ACNNs με  $r = 1$  και τα κελιά τους να έχουν μόνο "ενεργούς" γείτονες (δηλαδή μόνο τέσσερις συντελεστές μη-μηδενικούς στο  $A(i, j; k, l)$ ).
- Δεδομένα Τα κελιά ικανοποιούν την  $B(i, j; k, l) = 0, \forall i, j, k, l$  που υπάρχει ο συντελεστής ανάδρασης.
- Δομή Τα δίκτυα θα έχουν συμμετρικό template (οι μη-μηδενικοί συντελεστές του  $A(i, j; k, l)$  πρέπει να είναι ίσοι και διανεμημένοι διαγώνια

πχ,

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Γενικά οι παραπάνω συνθήκες δεν είναι ιδιαίτερα περιοριστικές και μπορούν να εφαρμοστούν εύκολα στα περισσότερα πρακτικά προβλήματα.

### 3.4.3 Εναλλακτικές Τεχνικές Ελέγχου

Μπορούμε να ελέγξουμε ένα *ACNN* με δύο τρόπους, ανάλογα με το αν πρέπει να ελεγχθεί όλο το δίκτυο αμέσως. Αναφερόμαστε σε ολικές και τοπικές μεθόδους.

Ένας ολικός έλεγχος βασίζεται στην επιτυχή εφαρμογή των *input* δεδομένων σε όλο το δίκτυο και την ερμηνεία των *output* καταστάσεων του *CNN*. Επομένως πρέπει να εφαρμοστεί ένα σύνολο προτύπων που εξαρτάται από το συγκεκριμένο δίκτυο. Η ώρα παραγωγής και η πολυπλοκότητα του ελέγχου αυξάνονται όσο αυξάνει και το μέγεθος του δικτύου και ο χρόνος που απαιτείται για την εφαρμογή των προτύπων θα εξαρτηθεί από τον αριθμό των διαφορετικών προτύπων καθώς και από το πόσο διαφορετικές είναι οι αρχικές συνθήκες.

Σε έναν τοπικό έλεγχο, κάθε κελί διαχωρίζεται από τα υπόλοιπα και ελέγχεται. Έρα το πρόβλημα διασπάται στο αν θα ελέγξουμε παράλληλα ή σειριακά τα κελιά σε ένα δίκτυο, αφού τα απομονώσουμε. Τα ίδια πρότυπα εφαρμόζονται σαν *inputs* σε όλα τα κελιά και στη συνέχεια μεταφράζονται τα *outputs*. Μπορεί να πρόκειται για την καλύτερη τακτική, διότι είναι γενική (δεν εξαρτάται από το συγκεκριμένο δίκτυο), ο αριθμός των προτύπων είναι πολύ μικρότερος και η διάστασή τους μειώνεται. Στα μειονεκτήματα αναφέρουμε την εισαγωγή επιπλέον εξαρτημάτων υπολογιστή προκειμένου να γίνουν όλα τα κελιά προσβάσιμα κατά τη φάση του ελέγχου.

### 3.4.4 Έλεγχος Κελιών

Ας θεωρήσουμε το ισοδύναμο κύκλωμα ενός κελιού σε ένα *ACNN* όπως φαίνεται στο Σχ. 3.18. Το κύκλωμα έχει πέντε στοιχεία: ένα γραμμικό πυκνωτή με θετική χωρητικότητα  $C$ , μια γραμμική αντίσταση με τιμή  $R_x$ , δυο **piecewise linear voltage-controlled** αντιστάσεις και μια **time-varying** πηγή με *output* που στην περίπτωσή μας δίνεται από τον τύπο

$$g(t) = A(i, j; k, l)u_{ykl}(t)$$

λόγω του περιορισμού μας για τα *inputs* ( $B(i, j; k, l) = 0, \forall i, j, k, l$ ).

Ας υποθέσουμε ότι, χρησιμοποιώντας διακόπτες, μπορούμε να αποσυνδέσουμε όλες τις συζεύξεις από τα γειτονικά του και προς το κελί  $C(i, j)$ ,  $u_{ykl} = 0, \forall k, l = i, j$ . Σε αυτήν την περίπτωση η εξίσωση

### Σχήμα 3.19: Ισοδύναμο Κύκλωμα

ισορροπίας μπορεί να γραφτεί ως εξής

$$C \frac{du_{xij}}{dt} = -0.5A(i, j; i, j) (|u_{xij} + 1| - |u_{xij} - 1|) + \frac{u_{xij}}{R_x}$$

Αν μπορούμε να επαληθεύσουμε μια αρχική συνθήκη ή, ισοδύναμα, να εφοδιάσουμε το κελί (μέσω ενός διακόπτη) με μια αρχική τιμή  $V_{0xij}$ , διατυπώνουμε το παρακάτω θεώρημα που μπορεί να χρησιμοποιηθεί σαν βάση οποιασδήποτε στρατηγικής ελέγχου.

#### **Θεώρημα 3.4.1** Θεώρημα Τοπικού Ελέγχου

Σε ένα ACNN, αν οι παράμετροι του κυκλώματος ικανοποιούν την

$$A(i, j; k, l) > \frac{1}{R_{xij}}$$

έχουμε τότε τις ακόλουθες ιδιότητες:

$$\lim_{t \rightarrow \infty} |u_{xij}| > 1$$

και είτε

$$\lim_{t \rightarrow \infty} u_{yij} = 1 \text{ αν } V_{0xij} > 0 \text{ με } 1 < i < M, 1 < j < N$$

είτε

$$\lim_{t \rightarrow \infty} u_{yij} = -1 \text{ αν } V_{0xij} < 0 \text{ με } 1 < i < M, 1 < j < N$$

**Παρατήρηση 3.4.1** Στην περίπτωση που εφαρμόζονται οι υποθέσεις μας, υπάρχουν τρία σημεία ισορροπίας στο κελί, τα δύο από αυτά είναι σταθερά. Το τρίτο θα βρισχεται ενδιάμεσα από τα **bias voltages** και κατά κάποιον τρόπο αποτελεί ένα χώρισμα για τα σταθερά σημεία. Το θεώρημα μας επιτρέπει να ελέγξουμε ένα κελί χρησιμοποιώντας δύο τιμές του  $V_{0xij}$ , μια θετική και μια αρνητική, εκ των οποίων καμία δεν απαιτείται να είναι ιδιαίτερα ακριβής, όπως προκύπτει από το θεώρημα.

### 3.4.5 Αρχιτεκτονική Στρατηγική

Μπορεί κανείς να προτείνει διαφορετικές μεθοδολογίες για εφαρμογή τεχνικών ελέγχου σε κάθε κελί. Εισάγουμε μια από αυτές, τα βήματα που εφαρμόζονται είναι τα εξής:

1. Ξανασχηματίζουμε το πλέγμα των κελιών έτσι ώστε να απομονώσουμε κάθε κελί
2. Διορθώνουμε τις αρχικές συνθήκες και "αναγκάζουμε" όλα τα outputs σε +1 (Local Test Theorem)
3. Επεξεργασία/Αποθήκευση (τοπικά) της output τιμής
4. Διορθώνουμε τις αρχικές συνθήκες και "αναγκάζουμε" όλα τα outputs σε -1
5. Επεξεργασία/Αποθήκευση (τοπικά) της output κατάστασης
6. Σύγκριση (ψηφιακή) μεταξύ των δύο output καταστάσεων
7. Ανάγνωση του συγκρινόμενου output
8. Μετάφραση της ικανότητας κάθε κελιού

Στο Σχήμα 3.19 φαίνεται η βασική δομή για αυτήν την περίπτωση. Ο πυρήνας είναι ένα επεξεργάσιμο κελί. Αρκετοί διακόπτες ενώνουν (ή διακόπτουν) τα εξωτερικά inputs έτσι ώστε να συνδέονται είτε με τον πυρήνα (κατά τη διάρκεια της λειτουργίας) είτε με το συγκριτή (κατά τη διάρκεια της δοκιμής). Η λειτουργία του συγκριτή είναι να παρέχει

### Σχήμα 3.20: Προτεινόμενο Σχήμα για το κελί προς δοκιμή

κάποιο σήμα που να δείχνει αν τα τέσσερα σήματα που προέρχονται από γειτονικά κελιά είναι ίσα ή όχι. Αυτό γίνεται και για τις δύο καταστάσεις ισορροπίας. Σε αυτήν την περίπτωση κάθε κελί "γνωρίζει" αν τα γειτονικά του είναι όλα λειτουργικά. Περαιτέρω μελέτη θα προμηθεύσει "χάρτη" προκειμένου να αποκλειστούν τα ελαττωματικά κελιά.

Ένα πραγματικό κύκλωμα θα εξαρτηθεί από την εφαρμογή που θα επιλέξουμε για τον πυρήνα. Κατά τις γνωστές εφαρμογές CNNs, το ρεύμα μεταφέρει πληροφορία. Ένας απλός τρόπος σχεδιασμού ενός ικανού συγκριτή φαίνεται στο Σχήμα 3.20.

Στο Σχήμα 3.21 βλέπουμε μια αναπαράσταση πολλαπλασιαστή της βασικής δομής του 3.20. Αυτή η εικόνα αντιστοιχεί σε ένα από τα τμήματα του συγκριτή. Δύο συμπληρωματικές πηγές παρέχουν το σύνολο των inputs με τις διαφορετικές πολικότητες.

#### 3.4.6 Συμπεράσματα

Μελετήσαμε μια αρχιτεκτονική τέτοια ώστε να επαυξάνουμε την ικανότητα δοκιμής των ACNNs και να ανακουφίσουμε τη διαδικασία επα-

Σχήμα 3.21: Four - Neighbour current comparator (FNCC)

Σχήμα 3.22: Half Circuit for the FNCC

λήθευσης. Η μεθοδολογία αυτή είναι απλώς μια αρχική πρόταση προκειμένου να εκμεταλλευτούμε την ομαλότητα του δικτύου να πραγματοποιήσει έναν ολικό έλεγχο που μπορεί να ερμηνευτεί και επομένως έχει δυναμική εφαρμογή στον επανασχηματισμό του δικτύου. Η μέθοδος βασίζεται στη λογική του "διαίρει και βασίλευε" αφού στηρίζεται στην απομόνωση των κελιών και στη δοκιμή καθενός από αυτά με βάση τη διπλή ισορροπία αυτών, κατά την απομόνωσή τους. Απαιτεί επιπλέον hardware, επομένως πρέπει να επιτευχθεί κάποια ανταλλαγή όσον αφορά τη βελτίωση των δοκιμών σε σχέση με τη δύναμη, την κατανάλωση και την ταχύτητα.

## 3.5 ΕΦΑΡΜΟΓΕΣ

### 3.5.1 A CNN Handwritten Recognizer

Τα CNNs χρησιμοποιούνται για αναζήτηση χαρακτηριστικών στην αναγνώριση χειρόγραφων χαρακτήρων. Τα χαρακτηριστικά τροφοδοτούνται σε ένα απλό δίκτυο ταξινόμησης. Η απόδοση ελέγχθηκε χρησιμοποιώντας δύο γνωστές σειρές της βάσης δεδομένων ETL: Την ETL3 που αποτελείται από αριθμητικά, αλφάβητα και αρκετά σύμβολα και την ETL8B2 που αποτελείται από Ιαπωνικούς Hirakana χαρακτήρες. Το μέσο ποσοστό που πέτυχαν ήταν 94.8% και 85.7%, αντίστοιχα. Και οι δύο σειρές περιλαμβάνουν "δύσκολους" χαρακτήρες και αρκετά πειραγμένους, έτσι ώστε ακόμα και άνθρωπος δεν μπορεί να αναγνωρίσει.

#### Εισαγωγή

Σε κάθε πρόβλημα αναγνώρισης προτύπων, η αναζήτηση χαρακτηριστικών είναι πολύ σημαντική. Από τη στιγμή που έχουν εξαχθεί τα κύρια χαρακτηριστικά, ένα δίκτυο ταξινόμησης είναι εύκολο να κατασκευαστεί. Όταν η απόδοση όμως του δικτύου για την αναζήτηση χαρακτηριστικών είναι χαμηλή, είναι ιδιαίτερα δύσκολο, συχνά αδύνατο, να κατασκευαστεί ακριβής ταξινομητής. Σε αυτό δεν αποτελεί εξαίρεση η αναγνώριση χειρόγραφων προτύπων. Συγκεκριμένα, τα χαρακτηριστικά που εξάγουμε από χειρόγραφα πρότυπα, πρέπει να έχουν δύναμη (να παραμένουν δηλαδή σταθερά) απέναντι σε μεταβολές όπως: αλλαγή

θέσης, περιστροφή ή αλλαγή μεταξύ διαστάσεων. Ακόμα, τα υπολογισμένα αυτά χαρακτηριστικά πρέπει να έχουν πολύ μικρότερο "μέγεθος" από τα αρχικά δεδομένα. Υπάρχουν δύο διαφορετικά είδη ευστάθειας που πρέπει να διαχωριστούν: (α)γεωμετρική και (β)στατιστική.

Μιλώντας γενικά, η γεωμετρική αφορά τους τρεις τύπους μεταβολών που αναφέρθηκαν, ενώ η στατιστική αφορά άλλου είδους μεταβολές που μπορεί να προκληθούν. Πρέπει να τονιστεί ότι αυτές οι ευστάθειες είναι πολύ πιο δύσκολο να επιτευχθούν απ'ότι φαίνεται. Σε κάποιες περιπτώσεις, πρέπει να αποφεύγεται η απόλυτη ευστάθεια, γιατί αν ένα σύστημα παραμένει απολύτως σταθερό υπό περιστροφή, δεν θα μπορούμε να ξεχωρίσουμε το "6" από το "9", ή το "(" από το ")". Επομένως το να τροφοδοτούμε ένα, για παράδειγμα, *three-layer back-propagation* δίκτυο με ωμά δεδομένα και να ζητάμε να ταξινομήσει τους χαρακτήρες, είναι μη ρεαλιστικό. Θα δουλέψει μόνο για εντελώς απλοϊκά παραδείγματα. Υπάρχουν εκατοντάδες παραδείγματα αλγορίθμων που ανήκουν στο (α) και ακόμα περισσότερα που ανήκουν στο (β). Αυτό συνεπάγεται πως δεν έχει χαρακτηριστεί κάποιος αλγόριθμος ως ο καλύτερος. Το ίδιο ισχύει και για τους αλγορίθμους που προτείνονται στη συνέχεια. Θα ισχυριστούμε ότι το σύστημα που έχει επιλεχθεί για την αναγνώριση χειρόγραφων χαρακτήρων λειτουργεί.

Τα δεδομένα που χρησιμοποιήθηκαν ανήκουν στις ETL σειρές και περιέχουν γραφικούς χαρακτήρες από τουλάχιστον 100 διαφορετικά άτομα.

1. Η ETL3 αποτελείται από τους αριθμούς 1-9, το αγγλικό αλφάβητο (κεφαλαία) A-Z και 12 σύμβολα "-", "(", ")", "\*", "+", ",", "=", κλπ, κάποια από αυτά φαίνονται στο σχήμα (3.22)
2. Η ETL8B2 αποτελείται από 71 Γιαπωνέζικους χαρακτήρες Hirakana, κάποια παραδείγματα φαίνονται στο σχήμα (3.23)

Δύο παραδείγματα CNNs χρησιμοποιούνται στο σύστημα αναγνώρισης προτύπων που χρησιμοποιούμε προκειμένου να εξάγουμε τα απαραίτητα χαρακτηριστικά:

- *connected component detectors (CCDs)*
- *shadow detectors (SDs)*



Σχήμα 3.23: Τυπικά παραδείγματα αριθμών επιλεγμένα από τη βάση δεδομένων ETL3

Σχήμα 3.24: Τυπικά παραδείγματα χαρακτήρων Hirakana επιλεγμένων από τη βάση δεδομένων ETL8B2

Σχήμα 3.25: Συνδεδεμένα στοιχεία που έχουν εξαχθεί από τον αριθμό "5" προς τις τέσσερις κατευθύνσεις

*Τα διανύσματα που προκύπτουν εισάγονται σε ένα δίκτυο για ταξινόμηση. Από τα 100 δεδομένα για κάθε χαρακτήρα, 30% τυχαία επιλεγμένοι χαρακτήρες χρησιμοποιούνται για να εκπαιδεύσουν το δίκτυο.*

## **The Feature Extractors**

**Connected component detectors** *Τέσσερα διαφορετικά CCDs χρησιμοποιούνται για κάθε χαρακτήρα.*

*Η H- (αντίστοιχα V-) φόρμα εντοπίζει τον αριθμό των στοιχείων ενός δοσμένου διπολικού ( $\pm 1$ ) ειδώλου κατά την οριζόντια (κάθετη) κατεύθυνση. Όμοια, το D1- (ή D2- αντίστοιχα) κατά τη κατεύθυνση των  $45^\circ$  (ή  $-45^\circ$  αντίστοιχα). Στο σχήμα 3.24 βλέπουμε τα τέσσερα CCDs ενός τυπικού χαρακτήρα "5". Παρατηρούμε ότι και τα τέσσερα δείχνουν πλήρη ευστάθεια ως προς την αλλαγή θέσης. Η εμπειρία δείχνει ότι ως προς ένα ποσοστό έχουν και ευστάθεια ως προς την περιστροφή.*

**Shadow detectors** Παράδειγμα βλέπουμε στο σχήμα (3.25), όπου το *template* δίνει τη "σκιά" που προκύπτει όταν μια οριζόντια δέσμη φωτός εφαρμόζεται από τα δεξιά του χαρακτήρα.

**Data compression** Τα οκτώ χαρακτηριστικά που προέκυψαν στις προηγούμενες παραγράφους, δεν τροφοδοτούνται άμεσα σε δίκτυο απόφασης. Για να κατανοήσουμε το λόγο που συμβαίνει αυτό, ας θεωρήσουμε ότι μας έχει δοθεί ένα  $64 \times 64$  διπολικό πρότυπο. Τότε καθένα από τα οκτώ χαρακτηριστικά που προκύπτουν παραμένει ένα  $64 \times 64$  διπολικό πρότυπο.

### CCD data compression

Ας κοιτάξουμε το σχήμα (3.27a) που αποτελεί τυπικό παράδειγμα που προκύπτει από τη V-φόρμα του σχήματος (3.25). Αυτή η φόρμα δίνει εναλλακτικά  $\bullet$  και  $\circ$  σε κάθε στήλη, ο αριθμός των  $\bullet$  δείχνει τον αριθμό των συνδεδεμένων στοιχείων. Επομένως η πληροφορία εστιάζεται στο πως να εντοπίσουμε τα  $\circ$ . Στο συγκεκριμένο παράδειγμα υπάρχουν τέσσερα "επίπεδα" από  $\bullet$ . Προκειμένου να αποκωδικοποιήσουμε αυτά τα δεδομένα, ας ορίσουμε σαν  $Y$  το "πλάτος" του χαρακτήρα, πχ τον αριθμό των  $\bullet$  στο "χαμηλότερο" επίπεδο. Ορίζουμε

$$X_{11} = 0, \quad X_{12} = Y, \quad X_{13} = 0$$

Ο δείκτης "1" στο  $X_{1j}$  δηλώνει ότι τα δεδομένα του  $X_{1j}$  σχετίζονται με το χαμηλότερο επίπεδο. Από αυτά τα δεδομένα προκύπτει ο πίνακας (3.27b), που είναι ο πίνακας του  $X_{ij}$  και τον (3.27c), που είναι ο πίνακας του  $X_{ij}/Y$ . Ο (3.27d) δίνει τις αριθμητικές τιμές του (3.27b).

Εφόσον κάθε  $X_{ij}$  έχει διαιρεθεί με  $Y$ , τα αποτελέσματα έχουν μεγάλο ποσοστό ευστάθειας όσον αφορά την κλίμακα, κάτι που δεν έχουν τα ίδια τα CCDs. Αυτό θα το καλούμε V-δεδομένα. Αντίστοιχος χειρισμός μπορεί να γίνει στα δεδομένα που προκύπτουν από την H-φόρμα και στα αποτελέσματα θα αναφερόμαστε ως H-δεδομένα. Για να πάρουμε τα αντίστοιχα αποτελέσματα για τις D1- και D2-φόρμες, πρώτα ισιώνουμε\* τα δεδομένα (Σχήμα 3.28), κλίνοντας τμήμα του προτύπου κατά  $90^\circ$ . Τα αποτελέσματα είναι τα D1-δεδομένα και D2-δεδομένα αντίστοιχα.

Σχήμα 3.26: Σκιές που έχουν εξαχθεί από τον αριθμό "5" προς τις τέσσερις κατευθύνσεις

Σχήμα 3.27: Συμπύεση Δεδομένων: (a) τυπικό παράδειγμα CCD, όπου το  $X_{ij}$  καθορίζεται από τον αριθμό των  $\bullet$  ή των  $\circ$  στο  $i$  επίπεδο, (b)  $X_{ij}$ , (c)  $X_{ij}/Y$ , (d) οι τιμές του  $X_{ij}/Y$  για το (a)

### Σχήμα 3.28: "Ίσιώνοντας" τα δεδομένα των D1 και D2

Προκύπτει όμως ένα ερώτημα: πόσες παραλλαγές του  $X_{ij}/Y$  μπορούν να προκύψουν; Αν είναι του επιπέδου  $64 \times 64$  τα μετασχηματισμένα δεδομένα δεν έχουν βελτιωθεί καθόλου. Είναι όμως πολύ λιγότερα.

Ας θεωρήσουμε το Σχήμα (3.28a). Μόνο το κάτω αριστερό τμήμα (διαστάσεων  $10 \times 10$ ) του  $64 \times 64$  πίνακα φαίνεται. Τα υπόλοιπα είναι όλα ο. Σημειώνουμε πως τα συνδεδεμένα • δεν εννοούν απαραίτητα ότι το αρχικό πρότυπο έχει μόνο ένα συνδεδεμένο στοιχείο, αλλά ότι το  $X_{ij}/Y$  δεν είναι μηδενικό.

Το Σχήμα (3.29) δείχνει αποτελέσματα για την ETL8B2. Όπως είναι λογικό, απαιτεί περισσότερη πληροφορία αφού οι χαρακτήρες είναι πιο πολύπλοκοι από το λατινικό αλφάβητο και τα αριθμητικά.

**SD data compression** Παρατηρήσαμε ότι τα CCD είναι πολύ ικανά για αναγνώριση χειρόγραφων χαρακτήρων. Ένας από τους παράγοντες που μειώνει την αποδοτικότητά τους είναι η δυσκολία, για παράδειγμα, να ξεχωρίσουν την "(" από την ")" ή το "T" από το "Y" γιατί εμφανίζουν παρόμοια CCD δεδομένα. Ας εφαρμόσουμε την LS φόρμουλα (Σχήμα 3.29). Αμέσως βλέπουμε τη διαφορά. Φυσικά τα δεδομένα (3.29a) δεν περιέχουν καμία συμπίεση. Αν εφαρμόσουμε τη V-φόρμα σε αυτά τα δεδομένα, προκύπτει το Σχήμα (3.29b), όπου διακρίνεται

*p168*

Σχήμα 3.29: θέσεις των  $X_{ij}/Y$  με δεδομένα από την ETL8B2 (με  $\bullet$  συμβολίζονται τα μη-μηδενικά  $X_{ij}/Y$ ): (a) V-data, (b) H-data, (c) D1-data και (d) D2-data

Σχήμα 3.30: θέσεις των  $X_{ij}/Y$  με δεδομένα από την ETL3(με  $\bullet$  συμβολίζονται τα μη-μηδενικά  $X_{ij}/Y$ ): (a) V-data, (b) H-data, (c) D1-data και (d) D2-data



Σχήμα 3.31: SD Φόρμες που διακρίνουν το "( από το ")": (a) αποτελέσματα του LS για "( και )", (b) αποτελέσματα που προέκυψαν εφαρμόζοντας τη V-φόρμουλα στο (a)

Σχήμα 3.32: SE Φόρμες που διακρίνουν το "T" από το "Y"

καθαρά η "(" από την ")". Ομοίως στο Σχήμα (3.32) βλέπουμε πως τα *DS* δεδομένα μπορούν να χρησιμοποιηθούν για να διακρίνουμε το γράμμα "T" από το "Y".

**Το χαρακτηριστικό της αναλογίας** Προκειμένου να δυναμώσουμε τις δυνατότητες ανίχνευσης, προσθέτουμε ένα ακόμα χαρακτηριστικό, την αναλογία, που ορίζεται ως ο λόγος του μήκους του "πρώτου επιπέδου" των *H*-δεδομένων και των *V*-δεδομένων, αντίστοιχα. Ο πίνακας συνοψίζει τις διαστάσεις του διανύσματος που συγχωνεύει τα βασικά χαρακτηριστικά που χρησιμοποιήθηκαν σε αυτό το σύστημα αναγνώρισης χαρακτήρων, δηλαδή *CCD*, *SD* και την αναλογία των χαρακτηριστικών.

	<i>CCD</i>	<i>SD</i>	<i>Ratio</i>	Σύνολο
<i>ETL3</i>	57	8	1	66
<i>ETL8B2</i>	137	16	1	154

Σχήμα 3.33: Σχηματικό Διάγραμμα του συστήματος μας για αναγνώριση χειρόγραφων χαρακτήρων

## Το Δίκτυο Απόφασης

Ένα 3 - επιπέδων back - propagation δίκτυο (προσομοιωμένο σε ψηφιακό υπολογιστή), επιλέχθηκε ως δίκτυο απόφασης, απλώς επειδή ήταν άμεσα διαθέσιμο. Δεν υπήρχε ιδιαίτερος λόγος για την επιλογή αυτή και άλλα δίκτυα θα μπορούσαν να επιλεγθούν. Ο αριθμός των inputs είναι η διάσταση του διανύσματος που προέκυψε από τον προηγούμενο πίνακα (66 για ETL3 και 154 για ETL8B2). Ο αριθμός των outputs ισούται με τις κατηγορίες (χαρακτήρες) προς ταξινόμηση (48 για ETL3 και 154 για ETL8B2). Ο αριθμός των κρυφών μονάδων επιλέχθηκε να είναι ίσος με τον αριθμό των output μονάδων.

## Αποτελέσματα

Το διάγραμμα που χρησιμοποιήθηκε για το πείραμα είναι αυτό του Σχήματος 3.33. Τα δεδομένα που προμηθευτήκαμε από τις ETL σειρές είναι από πίνακα διαστάσεων  $64 \times 64$ , μερικά εκ των οποίων είναι ιδιαίτερα ακανόνιστα (βλ. Σχήμα 3.34).

**Πείραμα 1** Από τα 100 δεδομένα κάθε χαρακτήρα, 30% επιλέχθηκαν εντελώς τυχαία και χρησιμοποιήθηκαν για την εκπαίδευση του δικτύου. Το δίκτυο εκπαιδεύτηκε να βγάζει αποτέλεσμα +1 για κάθε επιθυμητή κατηγορία. Για έναν ALLIANT FX παράλληλο mini-supercomputer χρειάστηκαν περίπου 2 εβδομάδες για την ETL3 και 6 εβδομάδες για την ETL8B2. Μετά την εκπαίδευση όλα τα δεδομένα εμφανίστηκαν στο δίκτυο. Ο βαθμός αναγνώρισης εξετάστηκε με δύο διαφορετικά "thresholds":

1.  $threshold = 0.0$ , δηλαδή το output με τη μεγαλύτερη τιμή, όποια και αν είναι, θεωρείται ως απόφαση
2.  $threshold = 0.9$ , το δίκτυο θα αποφασίσει μόνο όταν υπερβεί το 0.9, διαφορετικά θεωρείται ως απόρριψη

Ο πίνακας που ακολουθεί δείχνει τα αποτελέσματα αυτού του πειράματος. Η απόρριψη θεωρείται σαν λάθος.

Σχήμα 3.34: Απλό δείγμα από κάποια δύσκολα δεδομένα της ETL, είναι δύσκολο να αναγνωριστούν ακόμα και από ανθρώπινο μάτι

	ETL3		ETL8B2	
Threshold	0.0	0.9	0.0	0.9
Average recognition Rate	94.8%	84.0%	85.7%	80.3%

**Πείραμα 2** Προκειμένου να ελέγξουμε την ποιότητα των **feature extractors**, τροφοδοτήσαμε το δίκτυο με όλα (100) τα δεδομένα για εκπαίδευση. Μετά την εκπαίδευση, τα δεδομένα αποκαλύφθηκαν στο σύστημα. Αν ο βαθμός αναγνώρισης ήταν 100%, οι ανιχνευτές μας θα ήταν τέλειοι. Όμως υπάρχει ένας ακόμη σημαντικός παράγοντας: μερικά αρχικά δεδομένα είναι τόσο "κακά" που ούτε άνθρωπος δεν μπορεί να ταξινομήσει. Στο Σχήμα 3.34 βλέπουμε κάποια. Έτσι ελέγξαμε το δίκτυο αφού αφαιρέσαμε αυτά τα δεδομένα (Ήταν 60 από  $100 \times 48$  για την ETL3 και 391 από  $100 \times 71$  για την ETL8B2). Ο Πίνακας δίνει τα αποτελέσματα. Δείχνει πόσο καλά λειτουργούν οι ανιχνευτές, σχεδόν σαν το ανθρώπινο μάτι.

	ETL3		ETL8B2	
Threshold	0.0	0.9	0.0	0.9
Average recognition Rate	99.9%	99.4%	99.9%	99.9%

**Χρόνος Εκτέλεσης** Όλα τα πειράματα έγιναν σε σε υπολογιστή με διπλή ακρίβεια.

- Η εξαγωγή χαρακτηριστικών (CNN) σε ψηφιακό σειριακό υπολογιστή, καθυστερεί διότι είναι σταθερό σημείο μιας 4096-διαστάσεων ODE. Αν το CNN κατασκευαστεί σε αναλογικό chip, η σύγκλιση καθορίζεται από τη χωρητικότητα του πυκνωτή και την αντίσταση της εφαρμογής και κρατάει κάποια msec.
- Η αποκωδικοποίηση του  $X_{ij}$  και το εμπρόσθιο κομμάτι του back-propagation δικτύου είναι απλά, αλλά μπορεί να απαιτούν αρκετά μεγάλη ακρίβεια. Για αυτόν το λόγο, αυτό το βήμα πρέπει να γίνει από ένα ψηφιακό σύστημα. Αν χρησιμοποιούνταν ένα DSP, θα ήταν θέμα msec.

- Η μάθηση θα απαιτούσε μεγάλη ακρίβεια και πολύ χρόνο όπως περιγράφηκε, κάτι που ισχύει στις περισσότερες back - propagation εφαρμογές. Επομένως αυτό το κομμάτι πρέπει να υλοποιηθεί σε ψηφιακό υπολογιστή.

### 3.5.2 Detecting Moving and Standing Objects Using CNNs

Θα μελετηθούν τέσσερα προβλήματα ανίχνευσης κίνησης. Το πιο απλό είναι ένα μοντέλο παρόμοιο με το περίφημο Hubel-Wiesel πείραμα με τον αμφιβληστροειδή μιας γάτας για τον εντοπισμό της κίνησης αντικειμένου με δοθείσα ταχύτητα και κατεύθυνση. Η πιο πολύπλοκη περίπτωση είναι ο προσδιορισμός της κάθετης και της οριζόντιας ταχύτητας ενός κινούμενου αντικειμένου. Διάφορες φόρμες χρησιμοποιούνται για τον εντοπισμό των διάφορων τύπων κίνησης. Στο παράδειγμα τα συνεχόμενα ασπρόμαυρα φωτογραφικά στιγμιότυπα τροφοδοτούν το input και τις αρχικές συνθήκες των κόμβων του cnn, αντίστοιχα.

#### Εισαγωγή

Ο εντοπισμός και η εκτίμηση της κίνησης αποτελούν ένα αειθαλές πρόβλημα. Αρκετοί αλγόριθμοι ψηφιακών υπολογιστών και πιο πρόσφατα αναλογικές "νευρωνικές" υπολογιστικές δομές έχουν χρησιμοποιηθεί για διάφορες κλάσεις τέτοιων προβλημάτων. Η περίφημη ανακάλυψη των Hubel και Wiesel απέδειξε πως αν μια μπάρα κινούταν επί μιας ορισμένης περιοχής του οπτικού πεδίου μιας γάτας και με συγκεκριμένη κατεύθυνση και αν η ταχύτητα ήταν κοντά σε μια δοθείσα τιμή, τότε κάποιοι συγκεκριμένοι (cortical) νευρώνες θα έδιναν σήμα (ανίχνευση).

Το CNN είναι ένα νέο και φυσικό πλαίσιο εργασίας για γεωμετρικά ομαλές, αναλογικές, μη-γραμμικές και δυναμικές υπολογιστικές δομές κατάλληλες ειδικά για εφαρμογές επεξεργασίας προτύπων. Θα παρουσιάσουμε κάποιους CNN αλγόριθμους (ακολουθίες από CNN templates), ικανούς να εντοπίσουν και να υπολογίσουν κάποιους τύπους κίνησης.

**Πρόβλημα 1** Ας θεωρήσουμε ένα απλό αντικείμενο (πχ μια μπάρα) που κινείται με σταθερή ταχύτητα. Σκοπός είναι η ανίχνευση του αντικειμένου αν κινείται με δοθείσα ταχύτητα και προς συγκεκριμένη κα-

τεύθυνση (έστω οριζόντια). Αυτό είναι παρόμοιο με το πείραμα των Hubel-Wiesel.

**Πρόβλημα 2** Σε αυτό το πρόβλημα η κατεύθυνση της κίνησης δε λαμβάνεται υπόψη. Σκοπός είναι ο εντοπισμός του αντικειμένου αν η απόλυτη τιμή της ταχύτητάς του ανήκει σε γνωστή κλίμακα τιμών (πχ ένα εικονοστοιχείο ανά περίοδο).

Σε αυτά τα δύο προβλήματα θα αναφερόμαστε ως προβλήματα ανίχνευσης κίνησης.

**Πρόβλημα 3** Προσδιορισμός της ταχύτητας αν το αντικείμενο κινείται οριζόντια.

**Πρόβλημα 4** Προσδιορισμός των στοιχείων  $u_x$  και  $u_y$  του διανύσματος της ταχύτητας, ενός αντικειμένου που κινείται αυθαίρετα.

Σε αυτά τα δύο προβλήματα θα αναφερόμαστε ως εκτιμήσεις ταχύτητας.

Μπορούν να θεωρηθούν δύο τύποι συνθηκών.

**Τύπος A** Το αντικείμενο είναι μαύρο ενώ το φόντο είναι λευκό. Το αντικείμενο είναι υποχρεωτικά άκαμπτο και **fully connected**, πρόκειται για την ιδανική συνθήκη.

**Τύπος B** Λαμβάνονται υπόψη διαταραχές, όπως **grey-scale** εικόνες, συνδετικότητα της εικόνας, συσκότιση και θόρυβος, διαταράξεις στην κατεύθυνση της κίνησης, πολλαπλά αντικείμενα, σκιές και αβεβαιότητα λόγω της πεπερασμένης ανάλυσης. Αυτή είναι μια πραγματική κατάσταση.

Οι δύο τύποι μοντέλων λειτουργίας είναι οι ακόλουθοι.

**Sampled mode** Οι συνεχόμενες ακολουθίες της κίνησης δίνονται ανά ίσα διαστήματα χρόνου.



**Continuous mode** Η ένταση των εικονοστοιχείων ποικίλλει συνεχώς στο χρόνο, επομένως απαιτεί τη χρήση **delay-type templates**.

**Σχόλια, γενικό πλαίσιο εργασίας και βασικές αρχές**

**Sampled Mode** Το κινούμενο αντικείμενο αναπαριστάται από μια σειρά δειγμάτων  $P_0, P_1, P_2, \dots$ . Ο χρόνος είναι  $\Delta t$ , επομένως  $P_q = P(q\Delta t)$ . Τα είδωλα κβαντικοποιούνται στο χώρο και απεικονίζονται σαν pixels διαστάσεων  $\Delta x \times \Delta y$ . Συνήθως  $\Delta y = \Delta x$ . Η ένταση του εικονοστοιχείου ποικίλλει από  $+1$  (μαύρο) έως  $-1$  (λευκό). Ας θεωρήσουμε δύο διαδοχικά είδωλα  $P(q\Delta t)$  και  $P((q+1)\Delta t)$ . Ένα είδωλο τροφοδοτείται σε ένα CNN (input) αναθέτοντας τις τιμές έντασης των εικονοστοιχείων του σε  $v_{uij}$  ή  $v_{xij}(0)$ , δηλαδή στους input κόμβους ή στις αρχικές καταστάσεις του δικτύου.

Χρησιμοποιώντας το κλασσικό CNN μοντέλο, η δυναμική ενός κελιού περιγράφεται από τις ακόλουθες εξισώσεις

$$\begin{aligned} C\dot{v}_{xij}(t) &= -R^{-1}v_{xij}(t) + \sum_{C(k,l) \in N_1(i,j)} A(i,j;k,l)v_{ukl} \\ &+ \sum_{C(k,l) \in N_1(i,j)} B(i,j;k,l)v_{ukl} + I, \quad 1 \leq i, \\ &1 \leq j, \quad k \leq M, \quad 1 \leq j, \quad l \leq N \end{aligned} \quad (3.47)$$

$$v_{xij}(t) = 0.5(|v_{xij}(t)+1|-|v_{xij}(t)-1|), \quad 1 \leq i \leq M, \quad 1 \leq j \leq N \quad (3.48)$$

$$|v_{xij}(0)| \leq 1 \quad \text{και} \quad |v_{uij}| \leq 1, \quad 1 \leq i \leq M, \quad 1 \leq j \leq N \quad (3.49)$$

όπου το  $N_1(i,j)$  δηλώνει την  $r=1$  γειτονιά του κελιού. Συνήθως με  $P(q\Delta t)$  τροφοδοτείται η input ηλεκτρική τάση ενώ με  $P((q+1)\Delta t)$  οι αρχικές καταστάσεις. Με την αποδοχή ότι  $q=0$ , θα θεωρούμε συχνά  $v_{uij} = P_0$  και  $v_{xij}(0) = P_1$ . Ο χρόνος τακτοποίησης του CNN δηλώνεται με  $t_s$ . Υποθέτουμε πάντα ότι  $\Delta t \gg t_s$ .

**Continuous mode** Το κινούμενο αντικείμενο εκφράζεται από το είδωλο  $P = P(t)$ , τις εντάσεις του εικονοστοιχείου που ποικίλλουν στο χρόνο. Αυτό το είδωλο τροφοδοτεί τους input κόμβους του CNN:  $v_{uij} = P(t)$ . Η αρχική κατάσταση του δικτύου είναι συνήθως όμοια με το input είδωλο σε χρόνο  $t=0$ :  $v_{xij}(0) = P(0)$ .

Η δυναμική του δικτύου περιγράφεται από την εξίσωση

$$\begin{aligned}
 C \dot{v}_{xij}(t) &= -R^{-1} v_{xij}(t) + \sum_{C(kl) \in N_1(i,j)} A(i,j;k,l) v_{ykl}(t) \\
 &+ \sum_{C(kl) \in N_1(i,j)} B(i,j;k,l) v_{ukl}(t) + \sum_{C(kl) \in N_1(i,j)} A^T(i,j;k,l) v_{ykl}(t - \tau) \\
 &+ \sum_{C(kl) \in N_1(i,j)} B^T(i,j;k,l) v_{ukl}(t - \tau) + I
 \end{aligned} \tag{3.50}$$

με  $1 \leq i, k \leq M, 1 \leq j, l \leq N$

όπου  $A^T(i,j;k,l$  και  $B^T(i,j;k,l$  καλούνται **delayed feedback** και **delayed control operators** αντίστοιχα.

Η σχέση ανάμεσα στο χρόνο καθυστέρησης  $\tau$  και χρόνο μετάβασης του δικτύου,  $t_s$ , είναι  $\tau \gg t_s$ , το δίκτυο πρέπει να είναι δηλαδή πολύ πιο γρήγορο από απεικονισμένη κίνηση. Ο χρόνος καθυστέρησης  $\tau$  πρέπει να ρυθμιστεί σύμφωνα με την ταχύτητα της κίνησης. Η output εξίσωση (3.48) και οι περιορισμοί (3.49) παραμένουν αμετάβλητες.

### Βασικές αρχές της στρατηγικής επίλυσης

#### 1. The difference principle

Δύο διαδοχικά είδωλα συνδυάζονται αν εκμεταλλευτεί κανείς το μοναδικό χαρακτηριστικό του CNN που παρέχει τη δυνατότητα ανάθεσης δύο ανεξάρτητων ειδώλων στους input κόμβους ( $v_{xij}$ ) και στους κόμβους των αρχικών καταστάσεων ( $v_{xij}(0)$ ). Σε πολλές περιπτώσεις η λογική διαφορά αυτών των δειγμάτων γεννιέται σαν ένα ενδιάμεσο βήμα ενός πιο πολύπλοκου αλγορίθμου.

#### 2. The delay principle

Όταν το input δείγμα μεταβάλλεται συνεχώς και έχει ανατεθεί στους input κόμβους του δικτύου, οι διαδοχικές φάσεις της κίνησης παράγονται από ένα delay-type template. Υπάρχει μια τριπλή αντιστοιχία μεταξύ (i) του πραγματικού input και του δείγματος, (ii) του delayed input δείγματος και του προηγούμενου δείγματος και (iii) του delay χρόνου  $\tau$  και του χρόνου δείγματος  $\Delta t$ .

#### 3. The multi-template principle

Η λύση εκτελείται εφαρμόζοντας αρκετά διαφορετικά templates,

το ένα μετά το άλλο, προκειμένου να επιλυθούν πολύπλοκα προβλήματα ανίχνευσης κίνησης και υπολογισμού. Από αλγοριθμική πλευρά ο ρόλος των *templates* είναι παρόμοιος με αυτόν των μακροεντολών που χρησιμοποιούνται στον προγραμματισμό. Από αυτήν την άποψη δημιουργείται ένας αναλογικός αλγόριθμος από τη χρήση συνεχόμενων αναλογικών μακροεντολών (*templates*). Πιθανώς, αρκετά απλά *templates* μπορούν να ενοποιηθούν σε ένα πολύπλοκο, πχ ο αναλογικός αλγόριθμος μπορεί να συντομευθεί εγχαθιστώντας κάποιες πιο πολύπλοκες μακροεντολές.

## Motion Detection In Sampled Mode

**Λύση του προβλήματος 1 (περίπτωση εξαρτόμενη από την κατεύθυνση)** Βασικός σκοπός είναι να προταθεί μια ακολουθία *templates* ικανή να ανιχνεύσει ένα αντικείμενο κινούμενο προς μια προκαθορισμένη (οριζόντια) κατεύθυνση και αν η απόλυτη τιμή του διανύσματος της ταχύτητας βρίσκεται σε μια μικρή γειτονιά του  $V = \Delta x / \Delta t$ . Αν η κίνηση δεν πληροί κάποια από αυτές τις δύο απαιτήσεις, η οθόνη θα γίνει κατάλευκη, δεν θα έχει συμβεί ανίχνευση. Σε καταφατική περίπτωση η διαφορά των  $P_1$  και  $P_0$ , που ορίζεται ως  $P_d \triangleq P_1 \setminus P_0$ , θα παραμείνει στην οθόνη. Ο CNN αλγόριθμος αποτελείται από τρία διαδοχικά βήματα.

*Βήμα 1. Λαμβάνοντας τη διαφορά*

Προκειμένου να πάρουμε τη διαφορά μεταξύ των δύο στιγμιοτύπων  $P_1$  και  $P_0$ , χρησιμοποιείται ο πίνακας αλήθειας. Αυτή η λογική λειτουργία μπορεί να ερμηνευτεί με το ακόλουθο απλό *template*

$$A = [1 \cdot 1], \quad B = [-1], \quad I = -1$$

$$v_{uij} = P_0, \quad v_{xij}(0) = P_1, \quad v_{yij}(\infty) = P_d \quad (3.51)$$

Το αποτέλεσμα της διαφοράς απεικονίζεται στο σχήμα (3.35). Προφανώς, αν δεν υπάρχει κίνηση ή αν η κίνηση είναι πολύ αργή, η εικόνα της διαφοράς δεν θα περιέχει μαύρα εικονοστοιχεία, δηλαδή το πρόβλημα θα έχει επιλυθεί σε ένα βήμα. Το ερώτημα είναι το εξής: τι γίνεται αν η κίνηση είναι πολύ γρήγορη;

Σχήμα 3.35: (a) Το προηγούμενο δείγμα  $P_0$ . (b) Το τρέχον δείγμα  $P_1$ .  
(c) Η εικόνα της διαφοράς

$P_0$	$P_1$	$P_d$
$v_{uij}$	$v_{xij}(0)$	$v_{yij}(\infty)$
-1	-1	-1
-1	1	1
1	-1	-1
1	1	-1

Πίνακας I. Πίνακας αλήθειας για τη διαφορά

### Βήμα 2. Ανίχνευση ταχύτητας

Σε αυτό το βήμα θα καθοριστεί αν το εκτόπισμα ενός αντικειμένου κατά τη δειγματοληψία είναι μεγαλύτερο ή ίσο από ένα εικονοστοιχείο. Ας υποθέσουμε ότι το αντικείμενο κινείται οριζόντια. Από το σχήμα (3.35) μπορεί κανείς εύκολα να δει ότι αν το αντικείμενο κινηθεί δεξιά για ένα ακριβώς εικονοστοιχείο, τότε κάθε μαύρο εικονοστοιχείο του  $P_d$  θα έχει ένα αριστερό γειτονικό του, στο  $P_0$ . Αυτό σημαίνει πως πρέπει να σβήσουμε κάθε μαύρο εικονοστοιχείο στο  $P_d$  χωρίς αριστερό γειτονικό του στο  $P_0$ . Αυτή είναι η βασική ιδέα της άμεσης ανίχνευσης γείτονα. Το αντίστοιχο CNN template είναι το ακόλουθο

$$A = [0 \ 2 \cdot 1 \ 0], \quad B = [2 \ 0 \ 0], \quad I = -2v_{uij} = P_0,$$

$$v_{xij}(0) = P_d, \quad v_{yij}(\infty) = P_s \quad (3.52)$$

Η δράση του template μπορεί να φανεί στο σχήμα (3.36). Αυτήν τη φορά η κίνηση ήταν πολύ γρήγορη, επομένως η οθόνη έχει σβηστεί εντελώς.

Σχήμα 3.36: (a) Το προηγούμενο δείγμα  $P_0$ . (b) Το τρέχον δείγμα  $P_1$  που είναι το ίδιο με την εικόνα της διαφοράς  $P_1 = P_0$ . (c) άμεση ανίχνευση γειτονιάς. Το εκτόπισμα είναι μεγάλο, άρα η οθόνη θα ασπρίσει ως το τέλος της ταλάντωσης, δεν θα υπάρξει ανίχνευση. Τα γκριζα εικονοστοιχεία αναπαριστούν τη διαδικασία της ταλάντωσης

Σχήμα 3.37: (a) Το προηγούμενο δείγμα  $P_0$  (τυπωμένο σε γκρι εικονοστοιχεία) και το τρέχον  $P_1 = P_d$  (τυπωμένο με μαύρα). (b) Ανακριβής ανίχνευση ταχύτητας λόγω των κοντινών άκρων. Το εκτόπισμα ήταν μεγαλύτερο από ένα εικονοστοιχείο, η οθόνη όμως δεν άδειασε. (c) Βελτιωμένη ανίχνευση ταχύτητας. Το νέο template διαγράφει πρώτα τα κρίσιμα εικονοστοιχεία κοντά στις άκρες του προηγούμενου δείγματος. Το αντικείμενο εξαφανίζεται ως το τέλος της ταλάντωσης

Σχήμα 3.38: (a) Το προηγούμενο δείγμα  $P_0$ . (b) Το τρέχον δείγμα  $P_1$ . (c) Η εικόνα της διαφοράς  $P_d$ . (d) Το διαταραγμένο αποτέλεσμα της οριζόντιας μετακίνησης. Το αντικείμενο κινείται σε μια διαγώνια κατεύθυνση, αλλά το template ανίχνευσης ταχύτητας δεν είναι ικανό να εξαφανίσει αυτό το είδος κίνησης. Εσφαλμένη ανίχνευση προκύπτει.

Αυτό το *template* φαίνεται ιδανικό για ανίχνευση ταχύτητας. Αν όμως θεωρήσουμε το παράδειγμα του σχήματος (3.37), αποδεικνύεται ότι πιθανώς να έχουμε ανεπιθύμητα αποτελέσματα αν το μέγεθος του αντικειμένου είναι μεγαλύτερο από ένα εικονοστοιχείο. Το γκριζο σχέδιο στο (3.37a) αναπαριστά το προηγούμενο δείγμα, το οποίο δε φαίνεται στην οθόνη. Παρόλο το μεγάλο οριζόντιο εκτόπισμα, μερικά εικονοστοιχεία στην εικόνα της διαφοράς δεν θα εξαφανιστούν επειδή έχουν ακόμα παρακείμενα μαύρα εικονοστοιχεία στο  $P_0$ . Υπό αυτές τις συνθήκες πρέπει να τροποποιήσουμε το *template*. Θα διαγράψουμε ένα μαύρο εικονοστοιχείο στο  $P_d$  αν δεν έχει αριστερό γειτονικό στο  $P_0$  ή αν έχει ένα μόνο και ένα στα δεξιά του αντίστοιχα στο  $P_0$ . Ακολουθεί το νέο *template*:

$$A = [1 \ 2 \cdot 1 \ -2], \quad B = [2 \ 0 \ 0], \quad I = -4$$

$$v_{uij} = P_0, \quad v_{xij}(0) = P_d, \quad v_{yij}(\infty) = P_{is} \quad (3.53)$$

Η συμπεριφορά του φαίνεται στο σχήμα (3.37c).

**Βήμα 3. Ανίχνευση Κατεύθυνσης**

Μέχρι τώρα έχουμε καταφέρει να διαγράψουμε το αντικείμενο αν η οριζόντια μετακίνηση είναι μικρότερη ή μεγαλύτερη από ένα εικονοστοιχείο ανά δειγματοληψία. Φυσικά, λανθασμένη ανίχνευση θα συμβεί αν

Σχήμα 3.39: (a) Κίνηση με σωστή ταχύτητα. Το προηγούμενο δείγμα είναι το μαύρο ενώ το τρέχον είναι το γκριζο. (b) Κίνηση με υψηλή ταχύτητα

το αντικείμενο εκτοπίσει επίσης τον εαυτό του κάθετα (σχήμα 3.38). Επομένως πρέπει να εκτελέσουμε το τρίτο βήμα του αλγορίθμου, με το ακόλουθο template, που καθαρίζει την οθόνη στην περίπτωση της κάθετης εκτόπισης:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 \cdot 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 0 \\ -1 & 1 & 0 \end{bmatrix}, \quad I = -1$$

$$v_{uij} = P_0, \quad v_{xij}(0) = P_{is}, \quad v_{yij}(\infty) = P_{det} \quad (3.54)$$

**Λύση του προβλήματος 2 (περίπτωση ανεξάρτητη από την κατεύθυνση)** Σε αυτήν την περίπτωση θεωρούμε ένα διαφορετικό τύπο προβλήματος, εξετάζουμε δηλαδή αν η απόλυτη τιμή της ταχύτητας είναι κατάλληλη, αν για παράδειγμα ανήκει στην ακτίνα γύρω από το  $V = \Delta x / \Delta t$ . Δεν μας ενδιαφέρει πλέον η κατεύθυνση της κίνησης. Αν η κίνηση είναι πιο αργή ή ταχύτερη από μια καθορισμένη  $V$ , τότε η οθόνη θα γίνει τελείως λευκή, το αντικείμενο δηλαδή δεν θα εντοπιστεί. Στην καταφατική περίπτωση το τρέχον δείγμα θα παραμείνει στην οθόνη. Ένα κυρτό αντικείμενο είναι περιορισμένο να είναι μεγαλύτερο από ένα εικονοστοιχείο σε κάθε διεύθυνση και δεν επιτρέπεται να περιέχει τρύπες.

Σχήμα 3.40: Η διαφορά στην ευνοϊκή (a) και στη δυσμενή (b) κίνηση

*Η μέθοδος αποτελείται από τέσσερα βήματα. Τα δύο παραδείγματα του σχήματος (3.39) θα μας βοηθήσουν να καταλάβουμε τον αναλογικό αλγόριθμο.*

*Βήμα 1. Λαμβάνοντας τη διαφορά*

*Είναι ολόιδιο με το Βήμα 1 της προηγούμενης παραγράφου. Τα αποτελέσματα μετά Τα αποτελέσματα της αφαίρεσης φαίνονται στο σχήμα (3.40).*

*Βήμα 2. Ανίχνευση ταχύτητας*

*Είναι προφανές πως δεν μπορούμε να χρησιμοποιήσουμε το προηγούμενο template διότι παράγει λανθασμένη απάντηση εάν η κίνηση δεν είναι εντελώς οριζόντια. Όμως από το σχήμα (3.40) μπορούμε εύκολα να δούμε ότι το κινούμενο αντικείμενο έχει σωστή ταχύτητα μόνο αν τα μαύρα τμήματα της διαφοράς έχουν πλάτος ένα εικονοστοιχείο. Αυτό σημαίνει ότι στην καταφατική περίπτωση κάθε μαύρο εικονοστοιχείο έχει το πολύ τρεις μαύρους γείτονες, ενώ στην αντίθετη περίπτωση υπάρχουν σίγουρα εικονοστοιχεία που έχουν τουλάχιστον τέσσερα γει-*



τονικά. Επομένως πρέπει να καθορίσουμε αν τα μαύρα εικονοστοιχεία του  $P_d$  έχουν περισσότερα από τρία γειτονικά ή όχι. Αυτό μπορεί να επιτευχθεί με το επόμενο *template*

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & -0.75 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{bmatrix}, \quad I = -2$$

$$v_{uij} = P_d, \quad v_{xij}(0) = P_d, \quad v_{yij}(\infty) = P_s \quad (3.55)$$

*Βήμα 3. Βελτιωμένη ανίχνευση ταχύτητας*

Μετά το Βήμα 2, η εικόνα της διαφοράς θα παραμείνει στην οθόνη όταν η ταχύτητα είναι ευνοϊκή, ενώ κάποια εικονοστοιχεία της θα εξαφανιστούν σε αντίθετη περίπτωση. Στο Βήμα 3 συμπληρώνουμε την ανίχνευση της ταχύτητας. Άρα η εικόνα της διαφοράς παραμένει ολόκληρη στην οθόνη αν η ταχύτητα είναι σωστή, ενώ εξαφανίζεται αν η ταχύτητα είναι πολύ μεγάλη. Το *template* που ακολουθεί θα επιτύχει τη βελτιωμένη ανίχνευση ταχύτητας:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} -1 & -1 & 1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \quad I = 0$$

$$v_{uij} = P_d, \quad v_{xij}(0) = P_s, \quad v_{yij}(\infty) = P_{is} \quad (3.56)$$

Αυτό το *template* εξαφανίζει ένα εικονοστοιχείο εάν έχει ένα γειτονικό μαύρο στο  $P_d$  αλλά δεν υπάρχει μαύρο στην ίδια θέση στο  $P_s$ . Έχει ενδιαφέρον να παρακολουθήσουμε την επίδραση της διασποράς που συμβαίνει όταν δύο ή περισσότερα διαδοχικά εικονοστοιχεία πρόκειται να διαγραφούν από την οθόνη (σχήμα 3.41).

*Βήμα 4. Ανακατασκευή εικόνας*

Μέχρι τώρα η εικόνα της διαφοράς παρέμενε στην οθόνη αν η κίνηση βρίσκεται στη σωστή εμβέλεια ταχύτητας. Προκειμένου να ολοκληρώσουμε, χρησιμοποιούμε το ακόλουθο *template* που ανακατασκευάζει την

Σχήμα 3.41: (a) Το προηγούμενο δείγμα  $P_0$  (μαύρο) και το τρέχον  $P_1$ . (b) Η εικόνα της διαφοράς  $P_d$ . (c) Η διαδικασία εφαρμογής του template (3.56) επιτυγχάνει την ανίχνευση, διαγράφει τα εναπομείνοντα εικονοστοιχεία της εικόνας της διαφοράς. Στο πρώτο μέρος της **ταλάντωσης** τα εικονοστοιχεία που εξαφανίζονται είναι αυτά που έχουν ένα ή περισσότερα μαύρα γειτονικά που διαγράφηκαν στο βήμα 2 του αλγορίθμου. (d) Στο δεύτερο μέρος συμβαίνει μια υποτιθέμενη διασπορά

τρέχουσα εικόνα του κινούμενου αντικειμένου από την εικόνα της διαφοράς:

$$A = \begin{bmatrix} 0 \cdot 2 & 0 \cdot 2 & 0 \cdot 2 \\ 0 \cdot 2 & 0 \cdot 2 & 0 \cdot 2 \\ 0 \cdot 2 & 0 \cdot 2 & 0 \cdot 2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 0$$

$$v_{uij} = P_1, \quad v_{xij}(0) = P_d, \quad v_{yij}(\infty) = P_{det} \quad (3.57)$$

## MOTION DETECTION IN CONTINUOUS MODE

Ακολουθώντας τα ίχνη του αντικειμένου

Το απλούστερο πρόβλημα στο χώρο της ανίχνευσης κίνησης σε συνεχή χρόνο, είναι να ακολουθήσουμε τα ίχνη των μικρών αντικειμένων στο πεδίο όρασης. Αυτό μπορεί να λυθεί μέσω του παρακάτω απλού template

$$A = [1 \cdot 5], \quad B = [1 \cdot 1], \quad I = 1$$

$$v_{uij} = P(t), \quad v_{xij}(0) \quad (3.58)$$

Σχήμα 3.42: Η ανακατασκευή του τρέχοντος δείγματος του σχήματος (3.37a), χρησιμοποιώντας το βελτιωμένο ανιχνευτή ταχύτητας

Πρόκειται για ένα απλούστατο απόλυτα συμμετρικό, σταθερό *template* που ουσιαστικά εκτελεί τη δυαδική OR συνάρτηση. Υποθέτουμε πως το  $t_s$  (**transient time**) είναι μικρότερο από το χρονικό διάλειμμα μεταξύ δύο αλλαγών στην *input* εικόνα. Το *template* (3.58) "προσθέτει" κάθε καινούριο *input* πρότυπο στην πραγματική κατάσταση του CNN (σχήμα 3.43).

**Δημιουργώντας το κινούμενο τμήμα (εικόνα της διαφοράς)** Σε πολλά προβλήματα ανίχνευσης κίνησης το βασικό ερώτημα είναι πώς να διαχωρίσουμε τα κινούμενα τμήματα της εικόνας από αυτά που δεν μεταβάλλονται κατά τη διάρκεια του πειράματος. Το *template* (3.51) είναι ικανό να λύσει το πρόβλημα παίρνοντας τη λογική διαφορά των διαδοχικών δειγμάτων. Στην περίπτωση των συνεχών *input* εικόνων, δεν μπορούμε να χρησιμοποιήσουμε τους κόμβους των αρχικών καταστάσεων ως ανεξάρτητες *input* πύλες για το CNN. Προκειμένου να πάρουμε τη διαφορά μεταξύ της πραγματικής φάσης και κάποιας προηγούμενης φάσης της κίνησης, πρέπει να χρησιμοποιήσουμε το *delay-type template*:

$$A = [1], \quad B = [3], \quad I = [-3], \quad A_s = [0], \quad B_s = [-3]$$

$$\tau \geq 5RC, \quad v_{uij} = P(t), \quad v_{xij}(0) = P(0), \quad v_{yij}(t) = P_d(t) \quad (3.59)$$

Αυτό το απλό *template* αφαιρεί το αναβάλλον *input*  $P(t - \tau)$  από το πραγματικό  $P(t)$  και παρέχει την εικόνα της διαφοράς συνεχώς στο χρόνο.

Σχήμα 3.43: (a) Δύο σωματίδια που κινούνται δια μέσου της οθόνης. (b) Οι τροχιές των σωματιδίων. Το template (3.58) προσθέτει κάθε νέο στιγμιότυπο που έρχεται σαν input στην τροχιά που αναπαριστούν οι μεταβλητές των καταστάσεων

Σε μια πραγματική κατάσταση η παραγόμενη διαφορά θα είναι πειραγμένη από θόρυβο λόγω διάφορων αντανάκλασεων του φωτός και διαταραχών της input εικόνας (σχήμα (3.44)). Για να αποφευχθεί αυτή η δυσκολία, σχεδιάστηκε ένα νέο template και είναι ικανό να εκτελέσει αφαίρεση και φιλτράρισμα θορύβου ταυτοχρόνως.

Το template που ψάχνουμε είναι ανεξάρτητο της κατεύθυνσης, δείχνει τέλεια συμμετρία. Η δομή του είναι η ακόλουθη

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} a & a & a \\ a & b & a \\ a & a & a \end{bmatrix}, \quad I, \quad A^T = [0], \quad B^T = \begin{bmatrix} c & c & c \\ c & c & c \end{bmatrix} \quad (3.60)$$

όπου  $a, b, c, d$ , και  $I$  είναι σταθερές. Αυτό το template είναι εντελώς σταθερό και η τροχιά του κυκλώματος θα φτάσει τα **saturation regions**. Πώς υπολογίζονται όμως αυτές οι σταθερές;

Χρησιμοποιείται ο πίνακας αλήθειας  $I$  για τη λογική διαφορά. Σε αυτήν την περίπτωση η πρώτη στήλη ανήκει στο **delayed input**, η δεύτερη στο πραγματικό, ενώ η τρίτη παρέχει το αποτέλεσμα της ταλάντωσης του κυκλώματος. Για παράδειγμα, η πρώτη γραμμή του πίνακα  $I$  δηλώνει ότι εάν ένα συγκεκριμένο εικονοστοιχείο είναι λευκό στην πραγματική και στην **delayed input** εικόνα, η σταθερή κατάσταση της αντίστοιχης

μεταβλητής θα είναι  $-1$  ανεξάρτητα από τα γειτονικά κελιά. Σε αυτήν την περίπτωση ισχύει η ακόλουθη ανισότητα:

$$\dot{v}_{xij}(t) = -v_{xij}(t) + v_{yij}(t) + I - b \pm 8a - d \pm 8c < 0 \quad (0 < t < t_s) \quad (3.61)$$

Το σύμβολο  $\pm$  δηλώνει ότι αυτή η ανισότητα πρέπει να εκπληρωθεί ανεξάρτητα από τις input γειτονικές μεταβλητές (αν δηλαδή τα γειτονικά εικονοστοιχεία είναι λευκά ή μαύρα). Όμοια, έχουμε για την τρίτη και τέταρτη γραμμή του πίνακα  $I$

$$\dot{v}_{xij}(t) = -v_{xij}(t) + v_{yij}(t) + I - b \pm 8a + d \pm 8c < 0 \quad (0 < t < t_s) \quad (3.62)$$

$$\dot{v}_{xij}(t) = -v_{xij}(t) + v_{yij}(t) + I + b \pm 8a + d \pm 8c < 0 \quad (0 < t < t_s) \quad (3.63)$$

Η δεύτερη γραμμή του πίνακα απαιτεί επιπλέον μελέτη. Δηλώνει ότι αν ένα συγκεκριμένο εικονοστοιχείο είναι μαύρο στην πραγματική input εικόνα και το ίδιο είναι λευκό στην delayed, αυτό πρέπει να παραμείνει μαύρο. Αυτή θα ήταν η κανονική λειτουργία της αφαίρεσης. Ταυτόχρονα όμως, θέλουμε να φιλτράρουμε το θόρυβο στην εικόνα της διαφοράς. Αυτό σημαίνει ότι αν ένα εικονοστοιχείο έχει λιγότερα από δύο γειτονικά, θεωρείται θόρυβος και πρέπει να παραμείνει λευκό:

$$\dot{v}_{xij}(t) = -v_{xij}(t) + v_{yij}(t) + I + b + 8a - d + 8c < 0 \quad (0 < t < t_s) \quad (3.64)$$

$$\dot{v}_{xij}(t) = -v_{xij}(t) + v_{yij}(t) + I + b + 8a - d + 6c < 0 \quad (0 < t < t_s) \quad (3.65)$$

$$\dot{v}_{xij}(t) = -v_{xij}(t) + v_{yij}(t) + I + b + 8a - d + 4c > 0 \quad (0 < t < t_s) \quad (3.66)$$

Σύμφωνα με τη παρούσα μέθοδο, πρέπει να γράψουμε κάθε πιθανή δήλωση του πίνακα αλήθειας στη μορφή μιας ανισότητας. Φυσικά, οι ανισότητες πρέπει να είναι είτε αντισυγχρουόμενες, δεν θα μπορεί να

Σχήμα 3.44: (a) Διαφορετικά στιγμιότυπα ενός ρολογιού σε κίνηση. Το χέρι γυρίζει ενώ άλλα τμήματα της εικόνας είναι ακίνητα. Τα στιγμιότυπα τραβήχτηκαν με φωτογραφική μηχανή. Η ανάλυση της οθόνης είναι  $44 \times 44$ . (b) Το template (3.59) δίνει τη διαφορά συνεχώς στο χρόνο. Η output εικόνα έχει διαταραχές λόγω αντανάκλασεων του φωτός και μικρών διαταραχών του συστήματος.

σχεδιαστεί CNN να επιλύσει αυτό το πρόβλημα, ή υπεράριθμες, οπότε οι πλεονάζουσες μπορούν να παραληφθούν.

Αν θεωρήσουμε τις εξισώσεις (3.61) - (3.66) σε  $t = 0$ , έχουμε  $v_{xij}(0) = v_{yij}(0)$ . Αν οι ανισότητες ισχύουν για  $t = 0$ , θα ισχύουν και για  $0 < t < t_s$ , λόγω της μονοτονικότητας των **transients**. Επομένως λαμβάνουμε τις ακόλουθες ανισότητες:

$$\begin{aligned}
 I - b \pm 8a - d \pm 8c &< 0 \\
 I - b \pm 8a + d \pm 8c &< 0 \\
 I + b \pm 8a + d \pm 8c &< 0 \\
 I + b + 8a - d + 8c &< 0 \\
 I + b + 8a - d + 6c &< 0 \\
 I + b + 8a - d + 4c &< 0
 \end{aligned} \tag{3.67}$$

Αυτό το σύνολο καθορίζει ένα κυρτό πολύεδρο στον 5-διάστατο παραμετρικό χώρο. Πρέπει να επιλέξουμε τιμές για τα  $a$ ,  $b$ ,  $c$ ,  $d$ , και  $I$  έτσι ώστε το αντίστοιχο σημείο στον παραμετρικό χώρο να βρίσκεται

Σχήμα 3.45: Το template (3.68) είναι ικανό να κάνει αφαίρεση και φιλτράρισμα θορύβου ταυτόχρονα. Η εικόνα δείχνει ένα στιγμιότυπο συνεχώς μεταβαλλόμενου output

μέσα στο πολύεδρο. Μια κατάλληλη επιλογή είναι  $a = 0.25, b = 2, c = -0.25, d = -2$  και  $I = -4.75$ . Το **delay-type template** θα ήταν

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 2 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}, \quad I = -4.75$$

$$A^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B^T = \begin{bmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & -2 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{bmatrix}, \quad \tau \geq 10RC$$

(3.68)

Η απόδοση του template φαίνεται στο σχήμα (3.43).

**Παρατήρηση 3.5.1** 1. Επιπλέον βήματα της ανάλυσης της κίνησης μπορούν να εκπληρωθούν από συνεχόμενα επίπεδα CNN χρησιμοποιώντας την εικόνα της διαφοράς σαν input.

2. Πρόσφατα, μια παραλλαγή αυτής της μεθόδου χρησιμοποιήθηκε επιτυχώς για τη δημιουργία ενός **delay-type template** που είναι ικανό να επιλύσει το Πρόβλημα 2 (ανίχνευση κίνησης ανεξάρτητα από την κατεύθυνση):

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = -2$$

$$A^\tau = \begin{bmatrix} 0.68 & 0.68 & 0.68 \\ 0.68 & 0.68 & 0.68 \\ 0.68 & 0.68 & 0.68 \end{bmatrix}, \quad B^\tau = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \tau \geq 10RC \quad (3.69)$$

### Motion Drastically Corrupted By Noise (MIMICRY)

Ένα από τα πιο εντυπωσιακά χαρακτηριστικά κάποιων ζωντανών οργανισμών (πχ. χαμαιλέοντες, χταπόδια κλπ) είναι η ικανότητα να αλλάζουν χρωματισμό προκειμένου να αναμειχθούν με το περιβάλλον, αφού πρόκειται για το συνηθέστερο τρόπο προστασίας απέναντι στα αρπακτικά (καμουφλάζ). Μόνο εφόσον υπάρξει κίνηση στον περιβάλλοντα χώρο, η ακίνητη αυτή μίμηση θα αποκαλύψει το κρυμμένο ζώο.

Εμπνευσμένοι από το καμουφλάζ στη φύση, ορίζουμε μια νέα κλάση προβλημάτων - μίμηση - για να προκαλέσουμε τα CNNs. Ας θεωρήσουμε ένα κρυμμένο ζώο που στέκεται μπροστά σε ένα φόντο με θόρυβο. Εφόσον η στατιστική ιδιότητα του ασαφούς προτύπου του ζώου είναι ίδια με του φόντου, κανείς δεν μπορεί να διαχωρίσει το ζώο από περιβάλλον. Όμως, αν ο θόρυβος μεταβάλλεται στατιστικά ενώ το πρότυπο του ζώου παραμένει αμετάβλητο, το ζώο μπορεί να διαχωριστεί από το περιβάλλον, υπολογίζοντας κατά μέσο όρο τις εικόνες σε μια επαρκώς μεγάλη περίοδο.

Για αυτό το λόγο χρησιμοποιούμε το *template* (3.70) με μηδενική γειτονιά, αγνοώντας κάθε πληροφορία σχετικά με το χώρο της εικόνας. Υποθέτουμε ότι η σταθερά του χρόνου για την ολοκλήρωση είναι αρκετά μεγάλη:

$$A = [1], \quad B = [\epsilon], \quad I = 0$$

$$v_{uij} = P(t), \quad v_{xij} = 0, \quad \epsilon = 0.06 \quad (3.70)$$

Τα πειράματα έγιναν αναπαράγοντας μια σειρά εικόνων με τυχαίες διαταραχές που μεταβάλλονται συνεχώς στο χρόνο και προσαρμόζοντας τα εικονοστοιχεία της κεντρικής περιοχής σε αυθαίρετες αλλά σταθερές



Σχήμα 3.46: Δύο διαδοχικά στιγμιότυπα του συνεχούς input προτύπου της μίμησης. Ένα "ζώο" με σταθερό πρότυπο κρύβεται στο διαρκώς μεταβαλλόμενο περιβάλλον. Χάρη στην ομοιομορφία της πυκνότητας των εικονοστοιχείων, το ζώο παραμένει δυσδιάκριτο

τιμές όσον αφορά την έντασή τους. Η κεντρική περιοχή με ένα σταθερό πρότυπο θα αναπαριστά το "ζώο", ενώ τα τυχαία μεταβαλλόμενες εξωτερικές περιοχές θα αναπαριστούν το διαταραγμένο περιβάλλον. Δύο στιγμιότυπα της αλληλουχίας των input προτύπων φαίνονται στο σχήμα (3.46). Παρατηρούμε πως κανένα στιγμιότυπο δεν μας βοηθάει ιδιαίτερα. Αν όμως χρησιμοποιήσουμε την (3.70), τότε το ζώο θα γίνει ορατό στο κέντρο της εικόνας (σχήμα (3.47)).

Σε ένα πιο περίπλοκο πείραμα η σύγκλιση των output μεταβλητών που ανήκουν στο ζώο μπορεί να επιταχυνθεί χρησιμοποιώντας μια μη-γραμμική ανάδραση για να μετρήσουμε πόσο κοντά είναι οι output μεταβλητές στις περιοχές κορεσμού. Το κατάλληλο μη-γραμμικό template είναι το ακόλουθο:

$$A = \begin{bmatrix} f(x) & f(x) & f(x) \\ f(x) & 0.97 & f(x) \\ f(x) & f(x) & f(x) \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 0$$

$$v_{u_{ij}} = P(t), \quad v_{x_{ij}}(0) = 0, \quad \epsilon = 0.06 \quad (3.71)$$

όπου η  $f(x)$  δηλώνει μια μη-γραμμική συνάρτηση του  $x = v_{x_{ij}} + v_{x_{kl}}$ . Η μη-γραμμική χαρακτηριστική φαίνεται στο σχήμα (3.46): Αυτό το template εξετάζει πλήρως την πληροφορία του χώρου που παίρνουμε από την τοπική γειτονιά.

Σχήμα 3.47: Δύο διαδοχικά στιγμιότυπα του συνεχούς output προτύπου του template (3.70). Η αναμενόμενη τιμή του θορύβου στο φόντο είναι μηδενική, άρα το φόντο παραμένει γκριζο μετά την ολοκλήρωση. Οι τιμές των input εικονοστοιχείων που ανήκουν στο ζώο είναι σταθερές, επομένως οι αντίστοιχες output μεταβλητές θα τείνουν είτε στο +1 (μαύρο) είτε στο -1 (λευκό)

**Σχόλια** Τα αποτελέσματα αυτού του κεφαλαίου μπορούν να εφαρμοστούν σε πολλούς τομείς, πχ στη ρομποτική. Η υλοποίηση του κυκλώματος θα μπορούσε να είναι ένας **hardware accelerator** ή ένα *VLSI* κύκλωμα. Στο τελευταίο, η ικανότητα προγραμματισμού είναι ιδιαίτερα κρίσιμη. Οι μέθοδοι που παρουσιάστηκαν είναι εν μέρει αναλυτικές και εν μέρει ευρετικές. Αν και οι κλάσεις των προβλημάτων απαιτούν διαφορετικές ακολουθίες από templates, κάποια συνηθισμένα βήματα είναι προφανή. Στα παραδείγματά μας, παρουσιάστηκαν κυρίως ασπρόμαυρα πρότυπα, αν και τα κυκλώματα λειτουργούν και σε γκριζα, τα οποία όμως απαιτούν περαιτέρω μελέτη.

### 3.5.3 Cellular Neural Network Design For Solving Specific Image-Processing Problems

#### Εισαγωγή

Μια από τις σημαντικότερες ιδιότητες του *CNN* είναι η σχετική ευκολία στην πραγματοποίηση *VLSI* κυκλώματος, λόγω της αρχιτεκτονικής του. Θα προσδιορίσουμε απλά κυκλώματα για επίλυση προβλημάτων επε-

Σχήμα 3.48: Τα λειτουργικά σημεία του κελιού  $C_{ij}$  (a, b, c) και η εξωτερική μη-γραμμική συνάρτηση στο template της μίμησης (d)

ξεργασίας σημάτων. Οι όροι

1. **Feature** Κάθε δυαδική συνάρτηση που ορίζεται σε ένα διδιάστατο πλέγμα
2. **Desired Feature** Ένα στοιχείο που μας ενδιαφέρει σε ένα σύνολο χαρακτηριστικών
3. **Feature Detection** Η λειτουργία παραγωγής πληροφορίας σχετικά με την παρουσία ή την απουσία ενός χαρακτηριστικού. Αυτή η πληροφορία αναπαριστάται από μια διαφορά τάσης ενός output κελιού ισορροπίας, με δυαδική τιμή  $\{-1, +1\}$ . Η ηλεκτρική τάση χαρακτηρίζεται ως υψηλή αν ισούται με  $+1$  και ως χαμηλή αν ισούται με  $-1$ . Η τάση ενός κελιού  $e_{ij}$   $V(y_{ij})$  μετά την επεξεργασία είναι υψηλή εάν και μόνο εάν το κελί αντιστοιχεί στο κεντρικό εικονοστοιχείο ενός desired feature.
4. **Shape** Ένα σύνολο συνδεδεμένων μαύρων εικονοστοιχείων. Θα αναφερόμαστε σε ένα εικονοστοιχείο ως μαύρο, όταν αντιστοιχεί σε υψηλή output ή input τάση. Με τον όρο συνδετικότητα εδώ εννοούμε, ότι στην 1 - γειτονιά κάθε εικονοστοιχείου ενός συγκεκριμένου συνόλου, υπάρχει τουλάχιστον ένα ακόμα στοιχείο που επίσης ανήκει σε αυτό το σύνολο.

Η δυναμική κάθε δικτύου περιγράφεται από την εξίσωση (state equation):

$$C \frac{dV_{x_{ij}}}{dt} = -\frac{1}{R_x} V_{x_{ij}} + I + \sum_{kl} A_{ij,kl} V_{y_{kl}} + \sum_{kl} B_{ij,kl} V_{u_{kl}}$$

Στα παραδείγματα θεωρούμε ότι ο βαθμός των  $A_{ij,kl}$  και  $B_{ij,kl}$  είναι της ίδιας τάξης με του  $\frac{1}{R_x}$  και ότι  $R_x = 1$ .

### Μία Στρατηγική Σχεδιασμού για το CNN ενός Επιπέδου

Το πρώτο βήμα στη διαδικασία σχεδιασμού του CNN είναι η εκτίμηση του επιπέδου πολυπλοκότητας του κυκλώματος, που θα πραγματοποιήσει τη συγκεκριμένη αποστολή. Σε περιπτώσεις όπου η συνάρτηση επεξεργασίας του σήματος μπορεί να θεωρηθεί σαν μη - επαναλαμβανόμενη,

μπορούμε να περιμένουμε μια δομή CNN ενός μόνο επιπέδου να είναι επαρκής. Η εφαρμογή μπορεί να απλουστευθεί αν η συνάρτηση που θα εκπληρωθεί από το κύκλωμα μπορεί να εκφραστεί ως εξής.

Εκπομπή συγκεκριμένων σημάτων στο δίκτυο αν μια συγκεκριμένη συνθήκη ικανοποιείται σύμφωνα με την *input* πληροφορία.

Αυτή η δήλωση προτείνει πως μια στρατηγική σχεδιασμού για το CNN ενός επιπέδου πρέπει να εκμεταλλευτεί τους ξεχωριστούς ρόλους των τελεστών ανάδρασης και τροφοδοσίας (**feedback και feedforward**). Η συνθήκη των τάσεων του *input* κελιού πρέπει να ελεγχθεί από τα στοιχεία του **B**, ενώ η διάδοση πρέπει να ελεγχθεί από κατάλληλη σύνδεση των *output* τάσεων του κελιού με τα στοιχεία  $A_{i,j,kl}$ .

## Εφαρμογή της Στρατηγικής Σχεδιασμού για Δυαδική Επεξεργασία Προτύπου

Πλήρης ανάκτηση πληροφορίας σε ένα τοπικά αλληλοσυνδεδεμένο κύκλωμα. Στόχος είναι να εξάγουμε τα συγκεκριμένα σχήματα από ένα *input* πρότυπο. Το αρχικό πρότυπο, που αποτελείται από τα εικονοστοιχεία που αντιστοιχούν στις αρχικές καταστάσεις του κελιού, περιέχει απομονωμένα μαύρα εικονοστοιχεία που καλούνται εικονοστοιχεία κλειδιά. Αν κάποιο από αυτά επικαλύψει κάποιο από τα σχήματα του *input* προτύπου, αυτό το σχήμα εξάγεται. Το κύκλωμα που προτάθηκε<sup>2</sup> απαιτεί μια δομή CNN με τρία επίπεδα. Ο αλγόριθμος όμως της λύσης αυτού του προβλήματος, μπορεί να σχηματιστεί σύμφωνα με το πρότυπο που παρουσιάστηκε, άρα μπορούμε να περιμένουμε ότι η αποστολή μπορεί να πραγματοποιηθεί σε ένα CNN ενός μόνο επιπέδου. Ας υποθέσουμε ότι όλες οι αρχικές καταστάσεις των κελιών ( $Vx_{ij}(0)$ ) ρυθμίζονται σε "-1" εκτός από τα κελιά που αντιστοιχούν στα εικονοστοιχεία κλειδιά, για τα οποία  $Vx_{ij}(0) = "+1"$ .

Ο σχεδιασμός του *template* τονίζει τον ξεχωριστό ρόλο των τελεστών: ο τελεστής ανάδρασης είναι υπεύθυνος για τη διάδοση ("διάχυση") των ισχυρών καταστάσεων του κελιού, επομένως οι καταχωρήσεις του είναι θετικές. Επίσης, εφόσον δεν υπάρχει προτίμηση στην κατεύθυνση της διάδοσης, ο τελεστής είναι συμμετρικός. Ο τελεστής τροφοδοσίας εξετάζει αν μια *input* τάση είναι ισχυρή και μόνο τότε

---

<sup>2</sup> Από τους Chua και Shi

επιτρέπει αύξηση στην κατάσταση ώστε να περιέχει ένα μη-μηδενικό στοιχείο στο κέντρο, που να είναι θετικό. Επομένως τα templates είναι της μορφής

$$A = \begin{bmatrix} \alpha & \alpha & \alpha \\ \alpha & \alpha_0 & \alpha \\ \alpha & \alpha & \alpha \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \beta_0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \alpha, \beta_0 > 0$$

Εφόσον έχουμε υποθέσει ότι οι τάσεις έχουν δυαδικές τιμές, οι απαιτήσεις που πρέπει να τηρηθούν προκειμένου να λάβουμε κατάλληλες τιμές για τα  $\alpha$ ,  $\alpha_0$  και  $\beta_0$ , έχουν τη μορφή γραμμικών συνδυασμών των άγνωστων παραμέτρων. Οι απαιτήσεις είναι οι ακόλουθες (το  $\alpha_0$  ισούται με  $\alpha_0 - R^{-1}$ ).

1. Η τάση της κατάστασης του κελιού αυξάνεται μόνο όταν

- Υψηλή αρχική τάση (κλειδί) αντιστοιχεί σε υψηλή input τάση, κάτι που συμβαίνει μόνο όταν το επιλεγμένο εικονοστοιχείο ανήκει στο σχήμα:

$$\beta_0 + a_0 - 8\alpha + I > 0$$

- Η αρχική τάση είναι χαμηλή αλλά το εικονοστοιχείο ανήκει στο επιλεγμένο σχήμα, άρα η κατάσταση του πρέπει να αυξάνει κατά τη διάρκεια της ταλάντωσης (**transient**):

$$\beta_0 - a_0 - 6\alpha + I > 0$$

2. Η τάση της κατάστασης του κελιού μειώνεται μόνο όταν

- Το εικονοστοιχείο δεν ανήκει στο επιλεγμένο σχήμα:

$$-\beta_0 - a_0 + 8\alpha + I < 0 \quad \text{ή} \quad \beta_0 - a_0 - 8\alpha + I < 0$$

- Ισχυρή αρχική κατάσταση αντιστοιχεί σε χαμηλή input τάση:

$$-\beta_0 + a_0 - 8\alpha + I > 0$$

Σχήμα 3.49: Ανάκτηση των συγκεκριμένων σχημάτων. (a) Η πληροφορία αποθηκευμένη στα inputs του CNN. Τα βέλη δείχνουν τα εικονοστοιχεία που υπερκαλύπτουν τα κλειδιά. (b) Αρχικές καταστάσεις του CNN (αρχικό πρότυπο με τα εικονοστοιχεία κλειδιά). (c) Το αποτέλεσμα της επεξεργασίας

Αυτό το σύνολο των ανισώσεων μπορεί να επιλυθεί με τεχνικές γραμμικού προγραμματισμού. Οι παράμετροι που χρησιμοποιήθηκαν στο παράδειγμα του σχήματος (3.49) είναι οι εξής:

$$\alpha_0 = 1.5, \quad \alpha = 0.3, \quad \beta_0 = 2.5, \quad I = 0$$

Πρέπει να τονιστεί ότι η πλήρης ανάκτηση πληροφορίας που μόλις περιγράφηκε, μπορεί να πραγματοποιηθεί χρησιμοποιώντας CNN ενός επιπέδου μόνο όταν τα κλειδιά είναι απομονωμένα. Στη γενική περίπτωση οι κατάλληλες απαιτήσεις συνιστούν ένα αλληλοσυγκρουόμενο σύνολο ανισώσεων.

**Extraction of shape feature** Ο στόχος μπορεί να περιγραφεί σαν η εξαγωγή όλων των εικονοστοιχείων που ανήκουν στο σχήμα που περιέχει το επιθυμητό χαρακτηριστικό. Αν θεωρήσουμε ότι για όλα τα κελιά ισχύει  $Vx_{ij}(0) = -1$ , τότε το κύκλωμα θα λειτουργούσε σύμφωνα με τον ακόλουθο κανόνα: τα κελιά που αντιμετωπίζουν το ζητούμενο χαρακτηριστικό πρέπει να αυξήσουν την κατάστασή τους και αυτή

η κατάσταση θα πρέπει να διαδοθεί και σε άλλα στοιχεία του σχήματος. Για άλλη μια φορά ο τελεστής ανάδρασης θεωρείται συμμετρικός. Ο τελεστής τροφοδοσίας (**feedforward**) θα ελέγχει αν ένα κελί ανταποκρίνεται στο ζητούμενο χαρακτηριστικό ή σε ένα σχήμα ενός input ειδώλου. Μια κατάλληλη μορφή των τελεστών είναι

$$A = \begin{bmatrix} \alpha & \alpha & \alpha \\ \alpha & \alpha_0 & \alpha \\ \alpha & \alpha & \alpha \end{bmatrix}, \quad B = \begin{bmatrix} \beta' & \beta' & \beta' \\ \beta' & \beta_0 & \beta' \\ \beta' & \beta' & \beta' \end{bmatrix}, \quad \beta' \neq 0, \quad \alpha > 0$$

$\beta' = \beta > 0$  αν το κελί που αντιστοιχεί στο κεντρικό εικονοστοιχείο του ζητούμενου χαρακτηριστικού περιμένει ένα μαύρο εικονοστοιχείο (πχ. υψηλή output τάση) στη συγκεκριμένη θέση και  $\beta' = -\beta$  διαφορετικά. Επιλέγοντα τιμές για τις παραμέτρους ώστε να ικανοποιούν τις ακόλουθες απαιτήσεις (θεωρούμε πάλι ότι  $a_0 = \alpha_0 - R^{-1} > 0$ ).

1. Η κατάσταση του κελιού αυξάνεται όταν

- ένα κελί αντιμετωπίζει ένα σχηματισμό των input τάσεων που αντιστοιχεί στο ζητούμενο χαρακτηριστικό:

$$\beta_0 + 8\beta - a_0 - 8\alpha + I > 0$$

- ένα κελί αντιστοιχεί στο σχήμα του input ειδώλου και έχει τουλάχιστον ένα γειτονικό σε υψηλή κατάσταση (διάδοση - **propagation**):

$$\beta_0 - 8\beta - a_0 - 6\alpha + I > 0$$

2. Η κατάσταση του κελιού δεν πρέπει να αυξηθεί όταν

- ένα κελί αντιμετωπίζει ένα χαρακτηριστικό παρόμοιο αλλά όχι πανομοιότυπο με το ζητούμενο:

$$\beta_0 + 6\beta - a_0 - 8\alpha + I < 0$$

- ένα εικονοστοιχείο δεν ανήκει σε κανένα σχήμα:

$$-\beta_0 + 8\beta - a_0 + 8\alpha + I < 0$$



### Σχήμα 3.50: Επανεξέταση του ζητούμενου χαρακτηριστικού

Αυτές οι απαιτήσεις δεν είναι αλληλοσυγκρουόμενες εάν υποθέσουμε ότι το κεντρικό εικονοστοιχείο του ζητούμενου χαρακτηριστικού είναι μαύρο. Όμως, μπορούμε πάντα να τροποποιήσουμε το περίγραμμα του ζητούμενου χαρακτηριστικού ώστε να μεγαλώσουμε τη γειτονιά του κελιού και να κάνουμε μαύρο το κεντρικό εικονοστοιχείο. Το βλέπουμε στο σχήμα (3.50) με τον αντίστοιχο **B**-πίνακα. Το παράδειγμα προς επεξεργασία με τις παραμέτρους που επιλέχθηκαν, παρουσιάζονται στο σχήμα (3.51).

### **CNN Cell Model Modification**

Υπάρχουν πολλά προβλήματα επεξεργασίας σημάτων που δεν μπορούν να επιλυθούν με ένα CNN ενός επιπέδου και πρέπει να χρησιμοποιήσουμε πιο πολύπλοκα κυκλώματα, πχ. CNNs πολλών επιπέδων. Όμως μερι-

Σχήμα 3.51: Εξαγωγή των σχημάτων που περιέχουν ένα ζητούμενο χαρακτηριστικό. (a) Το χαρακτηριστικό που μας ενδιαφέρει και το template. (b) Η input εικόνα. (c) Το αποτέλεσμα της επεξεργασίας

κές φορές το πρόβλημα μπορεί να αναμορφωθεί σε ένα σύνολο απλών εργασιών, με κάθε εργασία από αυτές να μπορεί να βγει εις πέρας χρησιμοποιώντας μια απλή δομή ενός επιπέδου. Οι εργασίες μπορούν να γίνουν παράλληλα, επομένως ως κατάλληλη αρχιτεκτονική κυκλώματος φαίνεται να είναι αρκετά CNNs σε παράλληλη χρήση. Ως παράλληλη σύνδεσή τους, κατανοούμε ένα κύκλωμα που αποτελείται από αρκετά CNNs, καθένα από τα οποία είναι σχεδιασμένο να εκτελεί διαφορετικές εργασίες, όπου κάθε CNN τροφοδοτείται με την ίδια input πληροφορία και οι αντίστοιχες output τάσεις κάθε δικτύου περνούν μέσα από τις κατάλληλες λογικές συναρτήσεις. Θεωρώντας τις δυσκολίες στην εφαρμογή μιας τέτοιας δομής σε VLSI, προτάθηκε μια ιδέα απλοποίησης του κυκλώματος. Σύμφωνα με αυτήν, αρκετά παράλληλα συνδεδεμένα CNNs αντικαθίστανται από ένα μόνο αποτελούμενο από τροποποιημένα κελιά. Το τροποποιημένο κύκλωμα επιτρέπει πολύπλοκη παράλληλη επεξεργασία σημάτων, ενώ το βασικό στοιχείο επεξεργασίας της προτεινόμενης δομής διαφέρει από το αρχικό κελί στο κομμάτι παραγωγής της κατάστασης. Συγκεκριμένα, η διαφορά δυναμικού της κατάστασης κάθε κελιού ελέγχεται από αρκετά υποκύκλωμα μέσω μιας κατάλληλης **diode-resistive** δομής που εφαρμόζει τη ζητούμενη **quasi-logical** λειτουργία.

Ο σχεδιασμός του δικτύου είναι το κομμάτι επιλογής του template για κάθε input υποκύκλωμα και η αρχιτεκτονική του ωμικού τμήματος

του κελιού. Κάθε  $\gamma$ -οστό input υποκύκλωμα περιγράφεται από παραμέτρους  $A_\gamma$ ,  $B_\gamma$  και  $I_\gamma$  και αυτές ορίζονται σαν να περιγράφουν ένα βασικό CNN που θα πραγματοποιούσε κάθε εργασία ξεχωριστά. Οι μερικές τάσεις ( $Vx_{ij}$ ) που παράγονται από αυτά τα υποκυκλώματα είναι η βάση για τον προσδιορισμό της κατάστασης του κελιού ( $Vx_{ij}$ ). Παρόλες τις λειτουργικές διαφορές ανάμεσα στην προτεινόμενη δομή και στα παράλληλα συνδεδεμένα CNNs, το αποτέλεσμα της επεξεργασίας σε πολλές περιπτώσεις είναι το ίδιο. Η τροποποιημένη δομή επίσης προσφέρει κάποιες χρήσιμες νέες ικανότητες επεξεργασίας.

### Binary Image - Processing Applications of CNNs Composed of Modified Cells

Ένα από τα προβλήματα για τα οποία μοιάζει κατάλληλη μια τροποποιημένη δομή είναι η ανίχνευση ενός συγκεκριμένου χαρακτηριστικού άσχετα με τον προσανατολισμό του στο input πρότυπο. Αυτό το πρόβλημα μπορεί εύκολα να διατυπωθεί σε μια μορφή που περιέχει απλές λειτουργίες που συνδέονται με λογικές εκφράσεις, για παράδειγμα πώς να εντοπιστεί το χαρακτηριστικό αν έχει ένα προσανατολισμό  $\phi$  OR αν έχει προσανατολισμό  $\psi$  OR... Το κύκλωμα του κελιού, οι τιμές των παραμέτρων και το παράδειγμα επεξεργασίας φαίνονται στο σχήμα (3.52). Το πρώτο input υποκύκλωμα του τροποποιημένου κελιού συμπεριφέρεται σαν ανιχνευτής του χαρακτηριστικού στον πρώτο του προσανατολισμό, ενώ ένα άλλο ανιχνεύει κάποιον άλλο προσανατολισμό. Ένα **diode - resistive** υποκύκλωμα παράγει την τάση της κατάστασης του κελιού, τη χαμηλότερη δηλαδή των τάσεων των input υποκυκλωμάτων.

Ένα από τα προβλήματα επεξεργασίας προτύπων που είναι δύσκολο αντιληφθούμε, είναι αυτό που παρουσιάζεται στο σχήμα (3.53). Ο στόχος - η εξαγωγή των σημείων του μέσου οριζόντιων γραμμών. - εξασφαλίζεται εκτελώντας την quasi - logical "AND" στις διαφορές δυναμικού που παράγονται από δύο input υποκυκλώματα σε κάθε κελί του δικτύου. Το πρώτο υποκύκλωμα δρα σαν ένα CCD (**connected component detector**) που μεταφράζει το πρότυπο κατά τη δεξιά άκρη του δικτύου, ενώ το δεύτερο είναι ένα CCD που δρα κατά την αντίθετη κατεύθυνση. Οι παράμετροι είναι οι ακόλουθες:

$$\mathbf{A1} = [1 \ 2 \ -1], \quad \mathbf{A2} = [-1 \ 2 \ 1], \quad \mathbf{B1} = 0, \quad \mathbf{B2} = 0, \quad I1 = 0, \quad I2 = 0$$

Σχήμα 3.52: Ανίχνευση χαρακτηριστικού χρησιμοποιώντας ένα CNN αποτελούμενο από τροποποιημένα κελιά. (a) Δομή του κελιού. (b) Το ζητούμενο χαρακτηριστικό στους δύο προσανατολισμούς και τα κατάλληλα templates. (c) Το αποτέλεσμα της επεξεργασίας.

Το κύκλωμα δουλεύει σωστά μόνο αν υπάρχει μονός αριθμός μαύρων στοιχείων στην οριζόντια γραμμή (σχήμα 3.53 (b)). Προκειμένου να πάρουμε έναν ανιχνευτή των μέσων σημείων χωρίς αυτόν τον περιορισμό, πρέπει να εισάγουμε ένα τρίτο input υποκύκλωμα που θα σπάσει τη συμμετρία του προβλήματος. Μια κατάλληλη επιλογή παραμέτρων για το επιπλέον τμήμα θα ήταν

$$\mathbf{A3} = [0 \quad -1 \quad 2 \quad 0 \quad -1], \quad \mathbf{B3} = 0, \quad I3 = -2 \cdot 1$$

Αν θεωρήσουμε τις διαφορές δυναμικού των output των κελιών των input υποκυκλωμάτων,  $Vx1_{ij}$ ,  $Vx2_{ij}$ ,  $Vx3_{ij}$  αντίστοιχα, τότε η διαφορά δυναμικού του σύνθετου κελιού παράγονται από τη σχέση

$$Vx_{ij} = \max\{Vx3_{ij}, \min\{Vx1_{ij}, Vx2_{ij}\}\}$$

που αντιστοιχεί στη λογική λειτουργία

$$Vx_{ij} = Vx3_{ij} + (Vx1_{ij} * Vx2_{ij})$$

όπου με "+" θεωρούμε τη λογικό OR και με "\*" το AND.

### Συμπεράσματα

Η μεθοδολογία που προτάθηκε, και δίνει έμφαση στους διαφορετικούς ρόλους των τελεστών  $\mathbf{A}$  και  $\mathbf{B}$ , είναι χρήσιμη στην ανάπτυξη νέων εφαρμογών για CNN ενός επιπέδου. Ακόμα, περιγράφοντας ένα πρόβλημα με βάση τις λειτουργίες που θα πραγματοποιηθούν στις input και output συνθήκες, μπορούμε να χρησιμοποιήσουμε τη στρατηγική σχεδιασμού για να κατανοήσουμε καλύτερα κάποιες υπάρχουσες εφαρμογές των CNN.

Παρουσιάστηκαν κάποιες ικανότητες στην επεξεργασία σημάτων ενός δικτύου με τροποποιημένα κελιά. Χρησιμοποιώντας πιο πολύπλοκα templates για τα input υποκυκλώματα, μπορούμε να αυξήσουμε τον αριθμό των εφαρμογών για αυτήν τη δομή, για παράδειγμα μπορούμε να εξάγουμε τα σχήματα που περιέχουν ένα από τα ζητούμενα χαρακτηριστικά, επέκταση δηλαδή της εφαρμογής που παρουσιάστηκε στην παράγραφο (3.5.2).

Σχήμα 3.53: Εξαγωγή των μέσων σημείων των οριζόντιων γραμμών. (a) Input πρότυπο. (b) Αποτέλεσμα της επεξεργασίας - κάθε CNN περιέχει δύο input υποκυκλώματα. (c) Δομή των κελιών με τρία input υποκυκλώματα.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9;147-169, 1985.
- [2] S.-I. Amari. Neural Theory of association and concept formation. *Biological Cybernetics*, 26: 175-185, 1977.
- [3] D. J. Amit, H. Gutfreund, and H. Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, 55: 1530-1533, 1985.
- [4] D. J. Amit, H. Gutfreund, and H. Sompolinsky. Information storage in neural networks with low levels of activity. *Physical Review, A* 35: 2293-2303, 1987.
- [5] J. A. Anderson. A memory storage model utilizing spatial correlation functions. *Kybernetik*, 5: 113-119, 1968.
- [6] J. A. Anderson. Two models for memory organization using interacting traces. *Mathematical Biosciences*, 8: 137-160, 1970.
- [7] J. A. Anderson. A simple neural network generating an interactive memory. *Mathematical Biosciences*, 14: 197-220, 1972.
- [8] J. A. Anderson. A theory for the recognition of items from short memorized lists. *Psychological Review*, 80: 417-438, 1973.
- [9] M. A. Cohen and S. Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 13: 815-826, 1983.

[10] T. Gszti. *Physical Models of Neural Networks*. World Scientific, Singapore, 1990.

[11] R. J. Glauber. Time-dependent statistics of the Ising model. *Journal of Mathematical Physics*, 4: 294-307, 1963.

[12] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York, 1949.

[13] G. E. Hinton and T. J. Sejnowski. Optimal perceptual inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 448-453, Washington, 1983. New York: IEEE Press.

[14] G. E. Hinton and T. J. Sejnowski. Learning and relearning in Boltzmann machines. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume I, pages 282-317. MIT Press, Cambridge MA, 1986.

[15] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences*, volume 79, pages 2554-2558, 1982.

[16] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-stated neurons. In *Proceedings of the National Academy of Sciences*, volume 81, pages 3088-3092, 1984.

[17] J. J. Hopfield and D. W. Tank. "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52: 141-152, 1985.

[18] T. Kohonen. Correlation associative memory. *IEEE Transactions on Computers*, C-21: 353-359, 1972.



[19] T. Kohonen. *Associative Memory - A System Theoretical Approach*. Springer-Verlag, New York, 1977.

[20] T. Kohonen and K. Ruohonen. Representation of associated data by matrix operations. *IEEE Transactions on Computers*, C-22: 701-702, 1973.

[21] W. A. Little and G. L. Shaw. Analytical study of the memory storage capacity of a neural network. *Mathematical Biosciences*, 39: 281-290, 1978.

[22] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5: 115-133, 1943.

[23] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh. The capacity of the Hopfield associative memory. *IEEE Transactions on Information Theory*, 33: 461-482, 1987.

[24] P. Peretto. On learning rule and memory storage abilities of asymmetrical neural networks. *Journal de Physique Lettres*, 49: 711-726, Paris, 1988.

[25] A. J. Ticknor and H. Barrett. Optical implementations of Boltzmann machines. *Optical Engineering*, 26: 16-21, 1987.

[26] W. Wee. Generalized inverse approach to adaptive multiclass pattern classifications. *IEEE Transactions on Computers*, C-17: 1157-1164, 1968.

[27] G. Weisbuch and F. Fogelman-Soulié. Scaling laws for the attractors of Hopfield networks. *Journal de Physique Lettres*, 46: 623-630, Paris, 1985.

[28] [McCulloch and Pitts, 1943] "A Logical Calculus of the Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics*, 5:115-133. Reprinted in [Anderson and Rosenfeld],

1988, pp. 18-28.

[29] [Anderson and Rosenfeld, 1988] *"Neurocomputing: Foundations of research.* Cambridge, MA: MIT Press

[30] [Pitts and McCulloch, 1947] *"How We Know Universals: The Perception of Auditory and Visual Forms."* **Bulletin of Mathematical Biophysics**, 9:127-147. Reprinted in [Anderson and Rosenfeld, 1988], pp. 32-42.

[31] [Hebb, 1949] *"The Organization of Behavior.* New York: John Wiley & Sons. Introduction and Chapter 4 reprinted in [Anderson and Rosenfeld, 1988], pp. 45-56.

[32] [Rochester, Holland, Haibt and Duda, 1956] *"Tests on a Cell Assembly Theory of the Action of the Brain, Using a Large Digital Computer"*. **IRE Transactions on Information Theory**, IT-2: 80-93. Reprinted in [Anderson and Rosenfeld, 1988], pp: 68-80.

[33] [McClelland and Rumelhart, 1988] *"Explorations in Parallel Distributed Processing"*. Cambridge, MA: MIT Press.

[34] [Von Neumann, 1958] *"The Computer and the Brain"*. New Haven: Yale University Press. Pages 66-82 are reprinted in [Anderson and Rosenfeld, 1988], pp. 83-89.

[35] [Block, 1962] *"The Perceptron: A Model for Brain Functioning, I"*. **Reviews of Modern Physics**, 34: 123-135. Reprinted in [Anderson and Rosenfeld, 1988], pp. 138-150.

[36] [Minsky and Papert, 1988] *"Perceptrons, Expanded Edition.* Cambridge, MA: MIT Press. Original edition, 1969.

[37] [Widrow and Hoff, 1960] *"Adaptive Switching Circuits"*. **IRE WESCON Convention Record**, part 4, pp. 96-104. Reprinted in [Anderson and Rosenfeld, 1988], pp. 126-134

[38] [Kohonen, 1982] "*Self-organized Formation of Topologically Correct Feature Maps*". *Biological Cybernetics*, **43**: 59-69. Reprinted in [Anderson and Rosenfeld, 1988], pp. 511-521.

[39] [Kohonen, Toerkkola, Shozakai, Kangas and Venta, 1987] "*Microprocessor Implementation of a Large Vocabulary Speech Recognizer and Phoenetic Typewriter for Finnish and Japanese*". *European Conference on Speech Technology, Edinburgh, September 1987. Volume 2*, pp. 377-180.

[40] [Kohonen, 1988] "*The "Neural" Phoenetic Typewriter*". *Computer*, **21(3)**: 11-22.

[41] [Angeniol, Vaubois and Le Texier, 1988] "*Self-organizing Feature Maps and the Travelling Salesman Problem*". *Neural Networks*, *1(4)*:289-293.

[42] [Anderson, 1968] "*A Memory Storage Model Utilizing Spatial Correlation Functions*". *Kybernetics*, **5**: 113-119. Reprinted in Anderson, Pellionisz and Rosenfeld, 1990, pp. 79-86.

[43] [Anderson, 1968] "*A Simple Neural Network Generating an Interactive Memory*". *Mathematical Biosciences*, **14**: 197-220. Reprinted in [Anderson and Rosenfeld, 1988], pp. 181-192.

[44] [Anderson, Silverstein, Ritz and Jones, 1977] "*Distinctive Features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model*". *Psychological Review*, **84**: 413-451. Reprinted in [Anderson and Rosenfeld, 1988], pp. 287-326.

[45] [Grossberg, 1976] "*Adaptive Pattern Classification and Universal Recoding, I: Parallel Development and Coding of Neural Feature Detectors*". *Biological Cybernetics*, **23**: 121-143. Reprinted in [Anderson and Rosenfeld, 1988], pp. 245-258.

[46] [Grossberg, 1980] "*How Does a Brain Build a Cognitive*

*Code?*" **Psychological Review**, **87**: 1-51. Reprinted in [Anderson and Rosenfeld, 1988], pp. 349-400.

[47] [Grossberg, 1982] *"Studies of Mind and Brain"*. **Boston: Reidel.**

[48] [Grossberg, 1987 and 1988] *The Adaptive Brain, I: Cognition, Learning, Reinforcement, and Rhythm, and II: Vision, Speech, Language, and Motor Control.* **Amsterdam: North-Holland.**

[49] [Carpenter and Grossberg, 1985] *"Category Learning and Adaptive Pattern Recognition, a Neural Network Model"*. **Proceedings of the Third Army Conference on Applied Mathematics and Computation, ARO Report 86-1, pp. 37-56.**

[50] [Carpenter and Grossberg, 1987a] *"A Massively Parallel Architecture for a Self-organizing Neural Pattern Recognition Machine."* **Computer Vision, Graphics, and Image Processing**, **37:54-115.**

[51] [Carpenter and Grossberg, 1987b] *"ART2: Self-organization of Stable Category Recognition Codes for Analog Input Patterns."* **Applied Optics**, **26**: 4919-4930.

[52] [Carpenter and Grossberg, 1990] *"ART3: Hierarchical Search Using Chemical Transmitters in Self-organizing Pattern Recognition Architectures."* **Neural Networks**, **3(4)**: 129-152.

[53] [Werbos, 1974] *"Beyond Regression: New Tools For Prediction and Analysis in the Behavioral Sciences (Ph.D.thesis).* **Cambridge, MA: Harvard U. Committee on Applied Mathematics.**

[54] [Bryson and Ho, 1969] *"Applied Optimal Control."* **New York: Blaisdell.**

[55] [Rumelhart, Hinton and Williams, 1986a] *"Learning In-*

*ternal Representations by Error Propagation.*" In D. E. Rumelhart and J. L. McClelland, eds., **Parallel Distributed Processing**, vol.1 chapter 8. Reprinted in [Anderson and Rosenfeld, 1988], pp. 675-695.

[56] [Rumelhart, Hinton and Williams, 1986b] "*Learning Representations by Back-Propagation Error.*" **Nature**, 323: 533-536. Reprinted in [Anderson and Rosenfeld, 1988], pp. 696-699.

[57] [McClelland and Rumelhart, 1988] *Explorations in Parallel Distributed Processing*. Cambridge, MA:MIT Press.

[58] [Hopfield, 1982] "*Neural Networks and Physical Systems with Emergent Collective Computational Abilities.*" **Proceeding of the National Academy of Scientists**, 79: 2554-2558. Reprinted in [Anderson and Rosenfeld, 1988], pp. 460-464.

[59] [Hopfield, 1984] "*Neurons with Graded Response Have Collective Computational Properties like Those of Two-state Neurons.*" **Proceedings of the National Academy of Sciences**, 81: 3088-2092. Reprinted in [Anderson and Rosenfeld, 1988], pp. 579-584.

[60] [Hopfield and Tank, 1985] "*Neural Computation of Decisions in Optimization Problems.*" **Biological Cybernetics**, 52: 141-152.

[61] [Hopfield and Tank, 1986] *Computing with Neural Circuits.*" **Science**, 233: 625-633.

[62] [Tank and Hopfield, 1987] "*Collective Computation in Neuronlike Circuits.*" **Scientific American**, 257: 104-114.

[63] [Kirkpatrick, Gelatt and Vecchi, 1983] "*Optimization by Simulated Annealing.*" **Science**, 220: 671-680. Reprinted in [Anderson and Rosenfeld, 1988], pp. 554-568.

[64] [Geman and Geman, 1984] "*Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*," **IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6: 721-741**. Reprinted in [Anderson and Rosenfeld, 1988], pp. 614-634.

[65] [Hinton and Sejnowski, 1985] "*Optimal Perceptual Inference*." **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington DC, pp., 448-453**.

[66] [Szu and Hartley, 1987] "*Fast Simulated Annealing*." **Physics Letters A, 122(3,4): 157-162**.

[67] [Farhat, Psaltis, Prata and Paek, 1985] "*Optical Implementation of the Hopfield Model*." **Applied Optics, 24: 1469-1475**. Reprinted in [Anderson and Rosenfeld, 1988], pp. 653-660.

[68] [Sivilatti, Mahowald and Mead, 1987] "*Real-time Visual Computations Using Analog CMOS Processing Arrays*." In P. Losleben, ed., **Advanced Research in VLSI: Proceedings of the 1987 Stanford Conference**. Cambridge, MA: MIT Press, pp. 295-321. Reprinted in [Anderson and Rosenfeld, 1988], pp. 703-712.

[69] [Johnson and Brown, 1988] "*Cognizers: Neural Networks and Machines that Think*." **New York: John Wiley & Sons**.

[70] [Hecht-Nielsen, 1990] "*Neurocomputing*." **Reading, MA: Addison-Wesley**.

[71] [Widrow and Hoff, 1960] **Widrow, B. and Hoff, M. E., Jr (1960) Adaptive switching circuits, IRE Western Electric Show and Convention Record, part 4, 96-140**.

[72] [Widrow and Stearns, 1985] **Widrow, B. and Stearns, S. D. (1985) Adaptive Signal Processing, Prentice Hall, Englewood**

Cliffs, N.J.